Introdução aos Problemas de Roteirização e Programação de Veículos

PNV-2450

André Bergsten Mendes

Problema de Programação de Veículos

Problema de Programação de Veículos

- Premissas
  - Os roteiros iniciam e terminam na base
  - Nem todos os veículos necessitam ser utilizados
  - A frota é fixa e homogênea
  - A demanda é conhecida e deverá ser atendida integralmente
  - O problema é de coleta (ou, de entrega)
  - **Há janelas de tempo no atendimento dos clientes**

Problema de Programação de Veículos

- Parâmetros - Conjuntos

$R = (N; A)$	Rede associada ao problema
$N = \{0,1,\ldots n\}$	Conjunto de nós (depósito + clientes)
$N^+ = N \setminus \{0\}$	Conjunto de nós (clientes)
$A = \{(i, j) : i, j \in N\}$	Conjunto de arcos
$V = \{0,1,\ldots m\}$	Conjunto de Veículos (índice k)

Problema de Programação de Veículos

- Parâmetros

c_{ij}	Custo de percorrer o arco (i,j)
Q	Capacidade de carga do veículo
d_i	Demanda do cliente i
$\tilde{t}_i$	Tempo de atendimento no cliente i
t_{ij}	Tempo de viagem entre i e j
T	Jornada de trabalho
a_i	Instante de abertuda da janela de tempo
b_i	Instante de encerramento da janela de tempo

Problema de Programação de Veículos

- Variáveis de Decisão

$$x_{ij}^{k} = \begin{cases} 1, \text{se o arco } (i,j) \in A \text{ for percorrido pelo} \\ \quad \text{veículo } k \in V \\ 0, \text{em caso contrário} \end{cases}$$

s_i Instante de início do atendimento do cliente i

Problema de Programação de Veículos

- Função Objetivo

$$\min C = \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij}^{k}$$

- Restrições

$$\sum_{\substack{i \in N \\ i \neq j}} \sum_{k \in V} x_{ij}^{k} = 1 \qquad \forall j \in N^{+}$$

$$\sum_{j \in N} x_{0j}^{k} = 1 \qquad \forall k \in V$$

Problema de Programação de Veículos

- Restrições

$$\sum_{i \in N} x_{i0}^{k} = 1 \qquad \forall k \in V$$

$$\sum_{\substack{i \in N \\ i \neq j}} x_{ij}^{k} = \sum_{\substack{h \in N \\ h \neq j}} x_{jh}^{k} \qquad \forall j \in N^{+}, \forall k \in V$$

$$\sum_{i \in N} d_i \left(\sum_{\substack{j \in N^{+} \\ i \neq j}} x_{ij}^{k} \right) \leq cap_k \qquad \forall k \in V$$

Problema de Programação de Veículos

- Restrições

$$s_j \geq s_i + \tilde{t}_i + t_{ij} - (1 - x_{ij}^k)T \qquad \forall i \in N, \forall j \in N^+, \forall k \in V$$

$$a_i \leq s_i \leq b_i \qquad \forall i \in N^+$$

$$x_{ij}^k \in \{0,1\} \forall (i,j) \in A, k \in V$$

$$s_i \geq 0 \qquad \forall i \in N^+$$

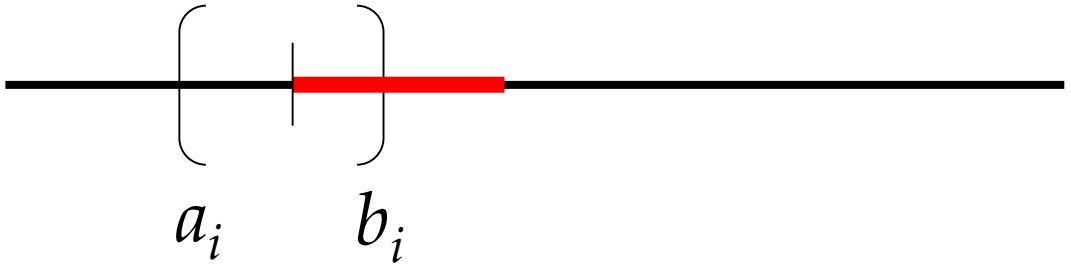
Problema de Programação de Veículos

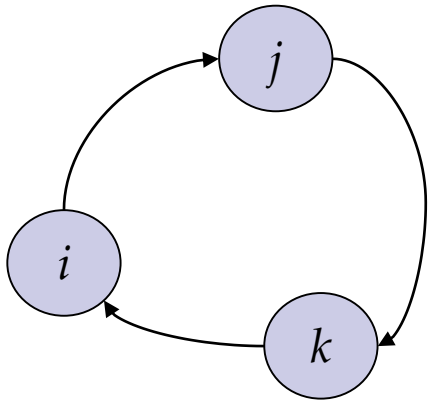
- Restrição de duração total do roteiro (jornada de trabalho)

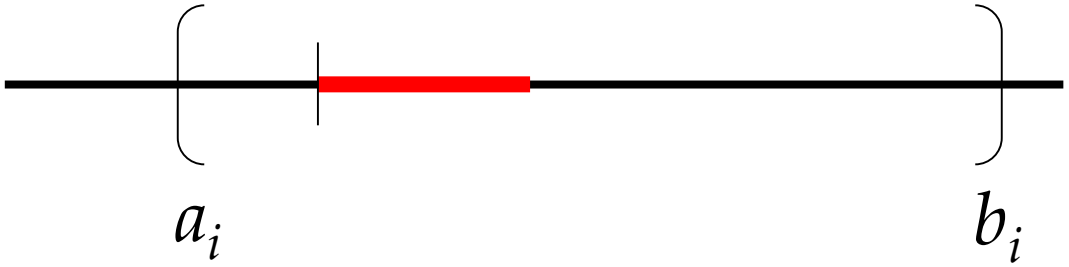
$$s_0 \leq T \qquad \forall k \in V$$

Problema de Programação de Veículos

Janelas de tempo & Progressão temporal







HEURÍSTICA DE INSERÇÃO DE SOLOMON (VRPTW)

Heurística de Inserção de Solomon

- Estratégia: construir uma solução por meio da inserção de clientes à rota, tal que os clientes já que já fazem parte da solução não tenham seus instantes de início modificados a ponto de violar suas janelas de tempo.

Heurística de Inserção de Solomon

- Parâmetros

s_i – tempo de atendimento no cliente i

e_i – instante inicial da janela de tempo do cliente i

l_i – instante final da janela de tempo do cliente i

t_{ij} – tempo de viagem entre os clientes i e j

d_{ij} – distância entre os clientes i e j

c_{ij} – custo de viagem entre os clientes i e j

Heurística de Inserção de Solomon

- Variáveis

  b_i – instante efetivo de início de atendimento do cliente i

  w_i – tempo de espera no cliente i

Heurística de Inserção de Solomon

- Definição de c_{ij}

  $c_{ij} = \rho_1 d_{ij} + \rho_2 (b_j - b_i) \qquad \rho_1 \geq 0, \rho_2 \geq 0$

- Instante de início do atendimento no cliente *j*

  $b_j = \max\{b_i + s_i + t_{ij}; e_j\}$

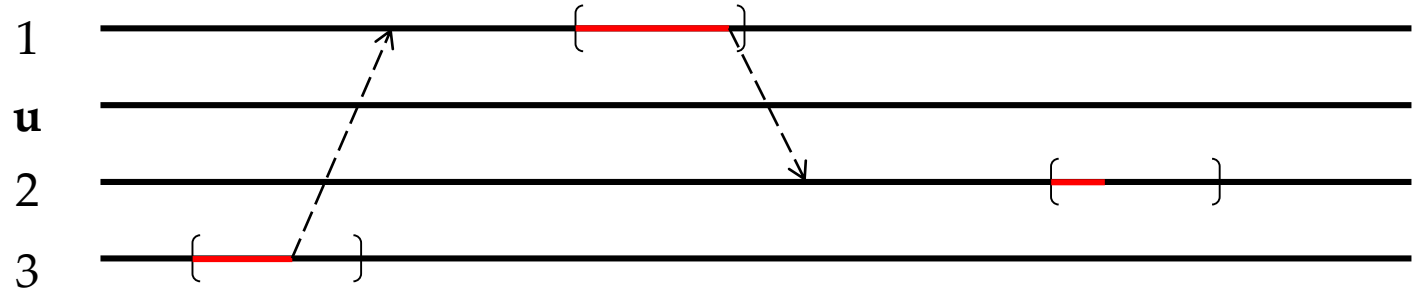
- Critério de viabilidade
  - Considere uma rota parcial viável $[i_0, i_1, \ldots, i_m]$ em que $i_0 = i_m = 0$
  - Avaliação da inserção do cliente *u* entre os clientes i_{p-1} e i_p

Heurística de Inserção de Solomon

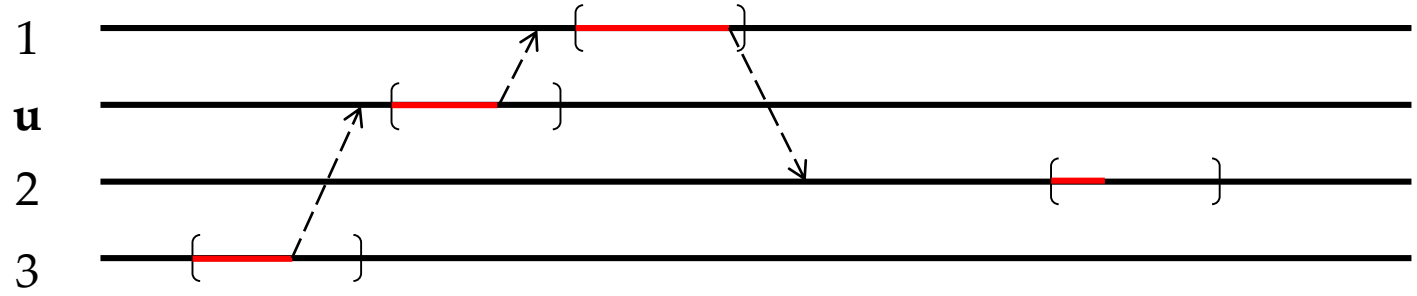
- Critério de viabilidade
  - A inserção do cliente u pode causar inviabilidade nos clientes i_p em diante
  - O instante efetivo de início de atendimento do cliente i_p em virtude da inserção do cliente u é: $b_{i_p}^{novo}$
  - O adiamento do início do atendimento do cliente i_p é: $PF_{i_p} = b_{i_p}^{novo} - b_{i_p} \geq 0$
  - Para os clientes finais da rota $PF_{i_{r+1}} = \max\{0, PF_{i_r} - w_{i_r}\} \qquad p \leq r \leq m-1$

Heurística de Inserção de Solomon

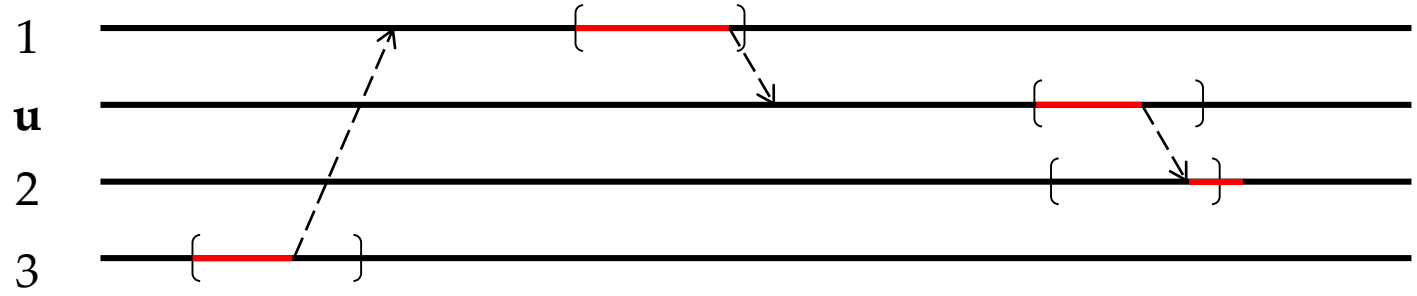
Rota Parcial



A inserção de **u** não interfere



A inserção de **u** atrasa o início de 2



Heurística de Inserção de Solomon

- Critério de viabilidade
  - As condições necessárias e suficientes para viabilidade da inserção do cliente u são:

    $b_u \leq l_u \qquad b_{i_r} + PF_{i_r} \leq l_{i_r} \qquad p \leq r \leq m$
- Critério de inicialização de uma rota
  - Escolher o cliente mais distante ainda não alocado ou o cliente não alocado cujo final da janela de tempo seja o menor de todos;

Heurística de Inserção de Solomon

- Critério de inserção
  - Considere rota parcial viável $[i_0, i_1, \ldots, i_m]$
  - Calcular para cliente u ainda não alocado, a melhor posição viável de inserção, definida por:

$$C_1(i(u), u, j(u)) = \min_p \{\mathbf{c_1}(i_{p-1}, u, i_p)\} \quad p = 1, \ldots, m$$

$$\mathbf{c_1}(i, u, j) = \alpha_1 c_{11}(i, u, j) + \alpha_2 c_{12}(i, u, j) \quad \text{onde :}$$

$$\begin{cases} \alpha_1 + \alpha_2 = 1 & \alpha_1 \geq 0, \alpha_2 \geq 0 \\ c_{11}(i, u, j) = d_{iu} + d_{uj} - \mu d_{ij} & \mu \geq 0 \\ c_{12}(i, u, j) = b_{j/u} - b_j \end{cases}$$

Heurística de Inserção de Solomon

- Critério de inserção: qual cliente u escolher?
  - Aplicar o critério

$$C_2\left(i(u^*),u^*,j(u^*)\right)=\max_{u}\left\{\mathbf{c}_2\left(i(u),u,j(u)\right)\right\} \quad \text{onde :}$$

$$\mathbf{c}_2\left(i,u,j\right)=\lambda d_{0u}-\mathbf{c}_1(i,u,j)$$

Exercício

- Para o problema de roteirização com janela de tempo abaixo indicado, gere uma solução por meio da Heurística de Inserção de Solomon. Adote os valores que julgar conveniente para $\alpha 1$, $\alpha 2$, μ e λ.

Exercício

- 7 clientes (os clientes 0 e 8 referem-se à base)
- A operação de distribuição inicia-se às 7hs, com os veículos já carregados, e é encerrada às 18hs
- Há, no máximo, 3 veículos homogêneos, com capacidade igual a 50 unidades de carga, cada
- As colunas “a” e “b” referem-se aos limites inferior e superior da janela de tempo para chegada aos clientes
- A coluna “Serviço” indica o tempo de atendimento do veículo junto a cada cliente

Exercício

Cliente	Demanda (unidades)	a (h)	b (h)	Serviço (h)
0	0	7	18	0
1	10	8	8,5	1
2	7	11,5	13	1
3	13	12	14	1
4	19	9	10	1
5	26	15	15,5	1
6	9	14	15	1
7	11	16	17	1
8	0	7	18	0

Matriz de Distância (km)

	0	1	2	3	4	5	6	7	8
0	0	15,2	18	22,3	25	20,6	11,1	21,2	0
1	15,2	0	32,5	14,5	32,2	32,2	24,8	21	15,2
2	18	32,5	0	34,4	20,2	23,8	16,4	36,2	18
3	22,3	14,5	34,4	0	25	42,7	33,5	35,3	22,3
4	25	32,2	20,2	25	0	41,2	31,6	46	25
5	20,6	32,2	23,8	42,7	41,2	0	10	20,6	20,6
6	11,1	24,8	16,4	33,5	31,6	10	0	20,6	11,1
7	21,2	21	36,2	35,3	46	20,6	20,6	0	21,2
8	0	15,2	18	22,3	25	20,6	11,1	21,2	0

Tempo de Deslocamento (h)

	0	1	2	3	4	5	6	7	8
0	0	0,7	0,9	1,1	1,2	1	0,5	1	0
1	0,7	0	1,6	0,7	1,6	1,6	1,2	1	0,7
2	0,9	1,6	0	1,7	1	1,1	0,8	1,8	0,9
3	1,1	0,7	1,7	0	1,2	2,1	1,6	1,7	1,1
4	1,2	1,6	1	1,2	0	2	1,5	2,3	1,2
5	1	1,6	1,1	2,1	2	0	0,5	1	1
6	0,5	1,2	0,8	1,6	1,5	0,5	0	1	0,5
7	1	1	1,8	1,7	2,3	1	1	0	1
8	0	0,7	0,9	1,1	1,2	1	0,5	1	0

Solução Ótima

Veículo	Rota				
1	0	4	2	8	
2	0	1	3	7	8
3	0	6	5	8	

Veículo	Instantes			
1	7	9	11,5	
2	7	8	12	16
3	7	14	15,5	

Distância total = 191,1

MÉTODOS DE BUSCA

Estratégias de Solução

- Busca Local
  - Visa a melhoria de uma solução, sem a garantia de otimalidade; requer a prévia definição da solução inicial que será explorada, da forma de geração da vizinhança, do critério de aceitação de uma solução gerada e de um critério de parada;
  - A geração da vizinhança ocorre em função do mecanismo empregado para criar novas soluções a partir da solução corrente como, por exemplo, a troca da posição de clientes e a substituição de arcos, entre outras;

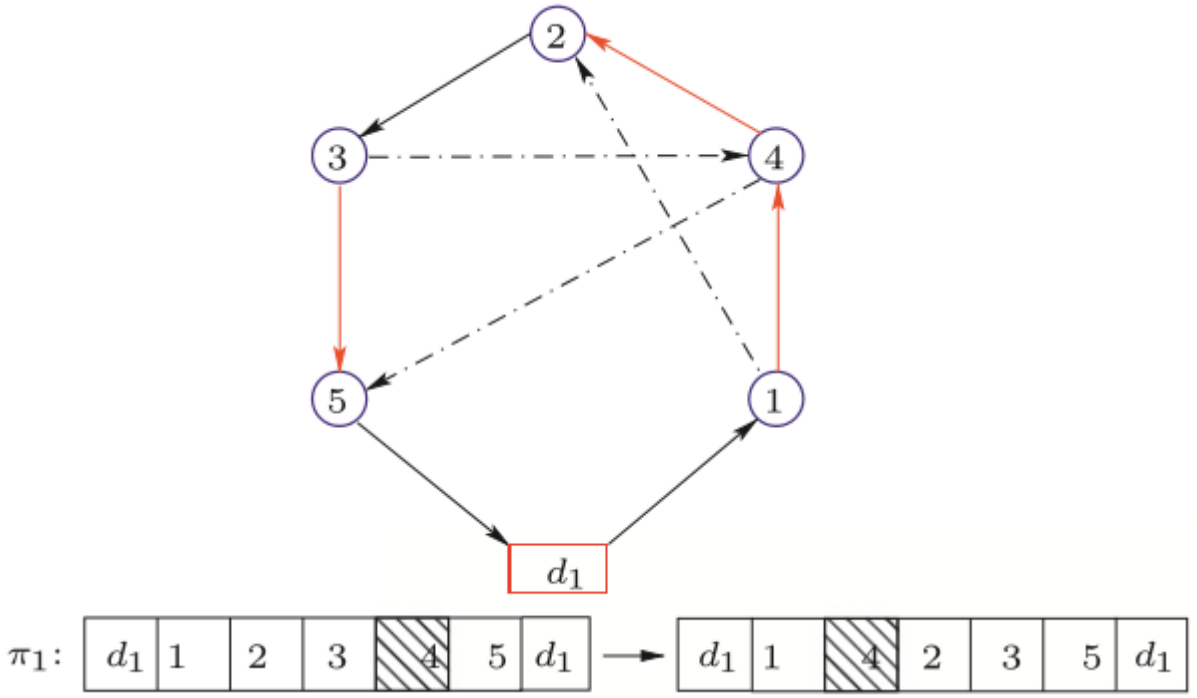
Estratégias de Solução

- Busca Local
  - Se a solução na vizinhança da solução corrente é melhor, então esta se torna a solução corrente, substituindo a anterior. Os critérios de aceitação comumente empregados são: 'escolhe-o-primeiro' ("first-accept") em que a primeira solução gerada melhor que a corrente é escolhida ou, 'escolhe-a-melhor' ("best-accept") de todas as soluções na vizinhança da solução corrente;
  - Caso não haja uma solução melhor, ter-se-á chegado a uma solução ótima local e o algoritmo encerra;

Operadores para geração de vizinhança

Remoção – Inserção (mesma rota)

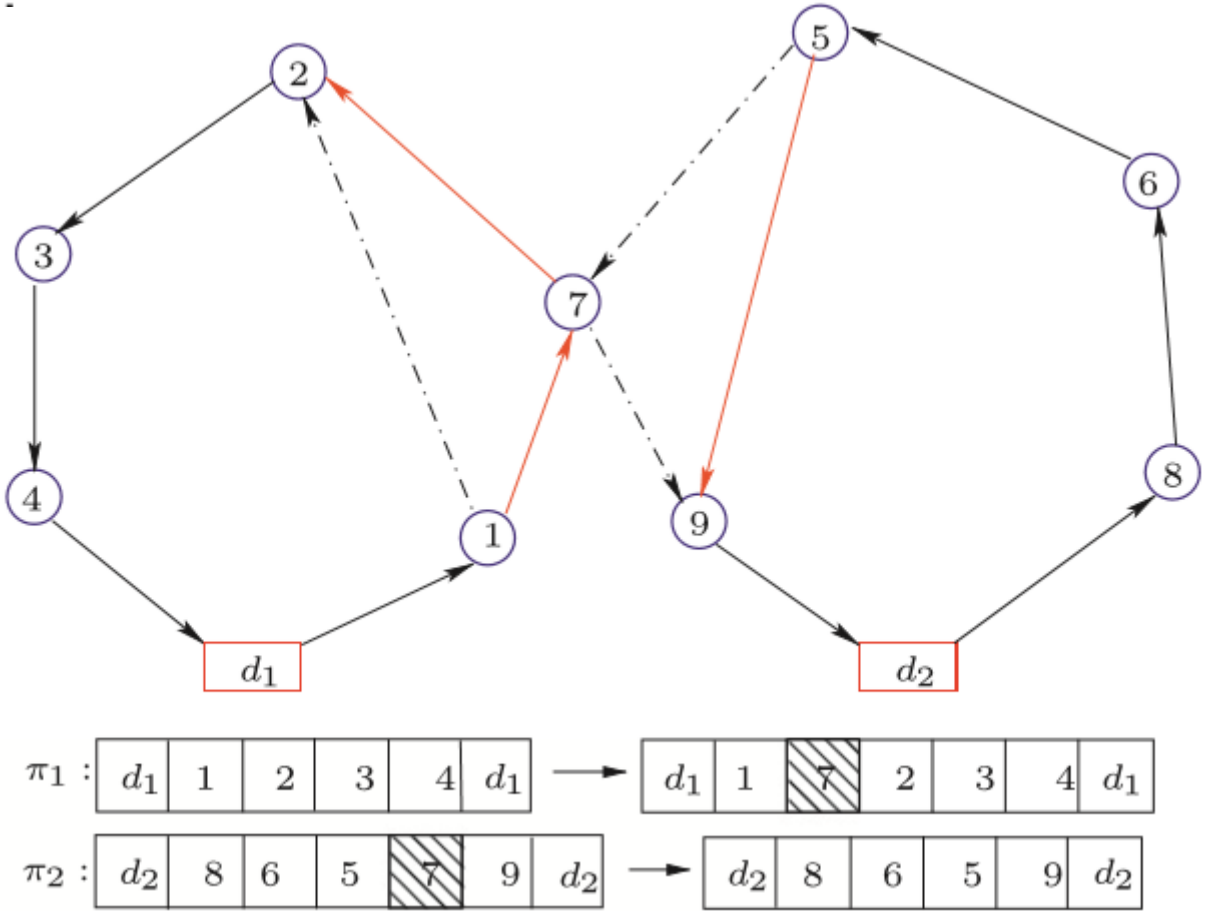
Jarboui et al. (2013)



Operadores para geração de vizinhança

Remoção – Inserção (entre rotas)

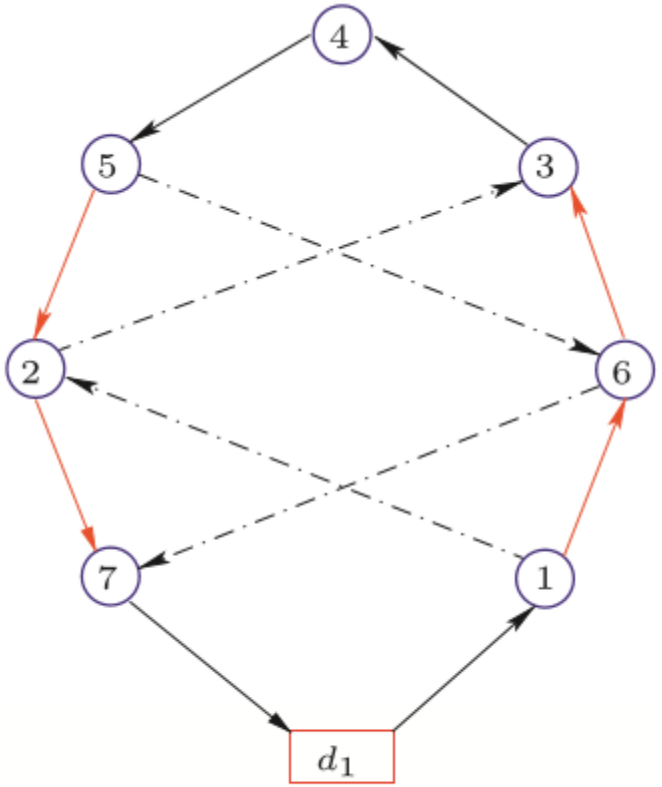
Jarboui et al. (2013)

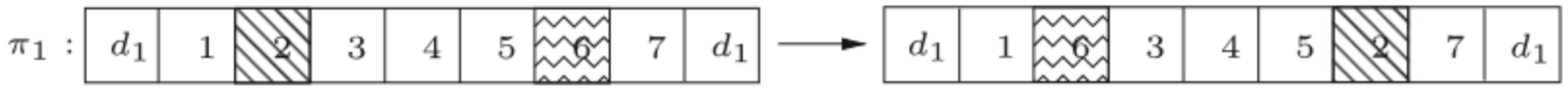


Operadores para geração de vizinhança

Swap (mesma rota)

Jarboui et al. (2013)

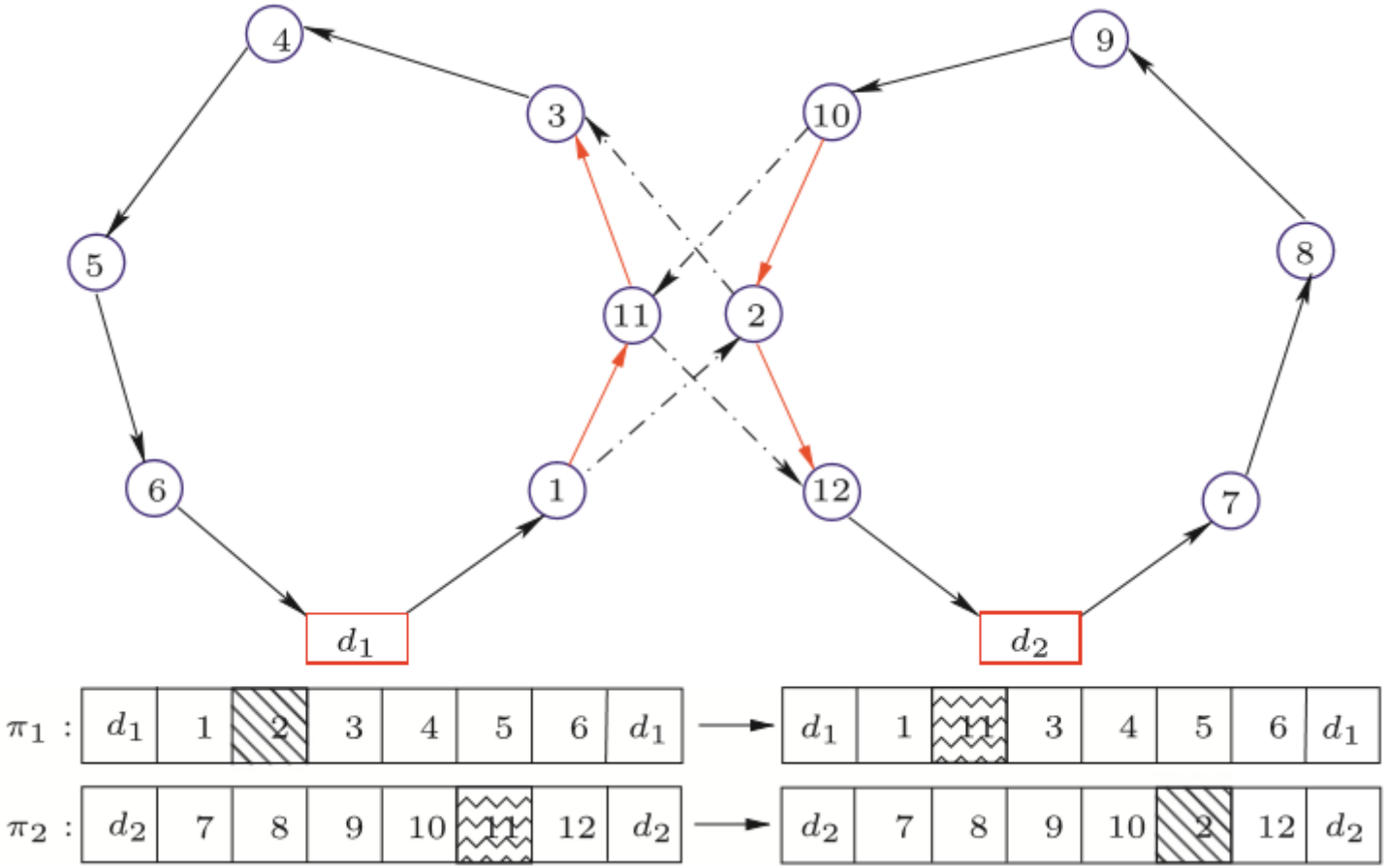




Operadores para geração de vizinhança

Swap (entre rotas)

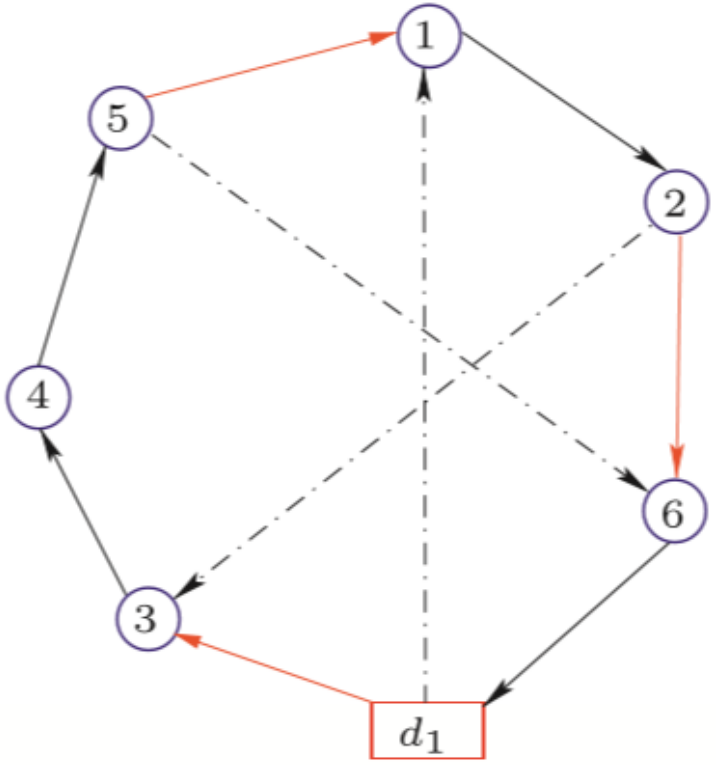
Jarboui et al. (2013)



Operadores para geração de vizinhança

Or-opt (mesma rota)

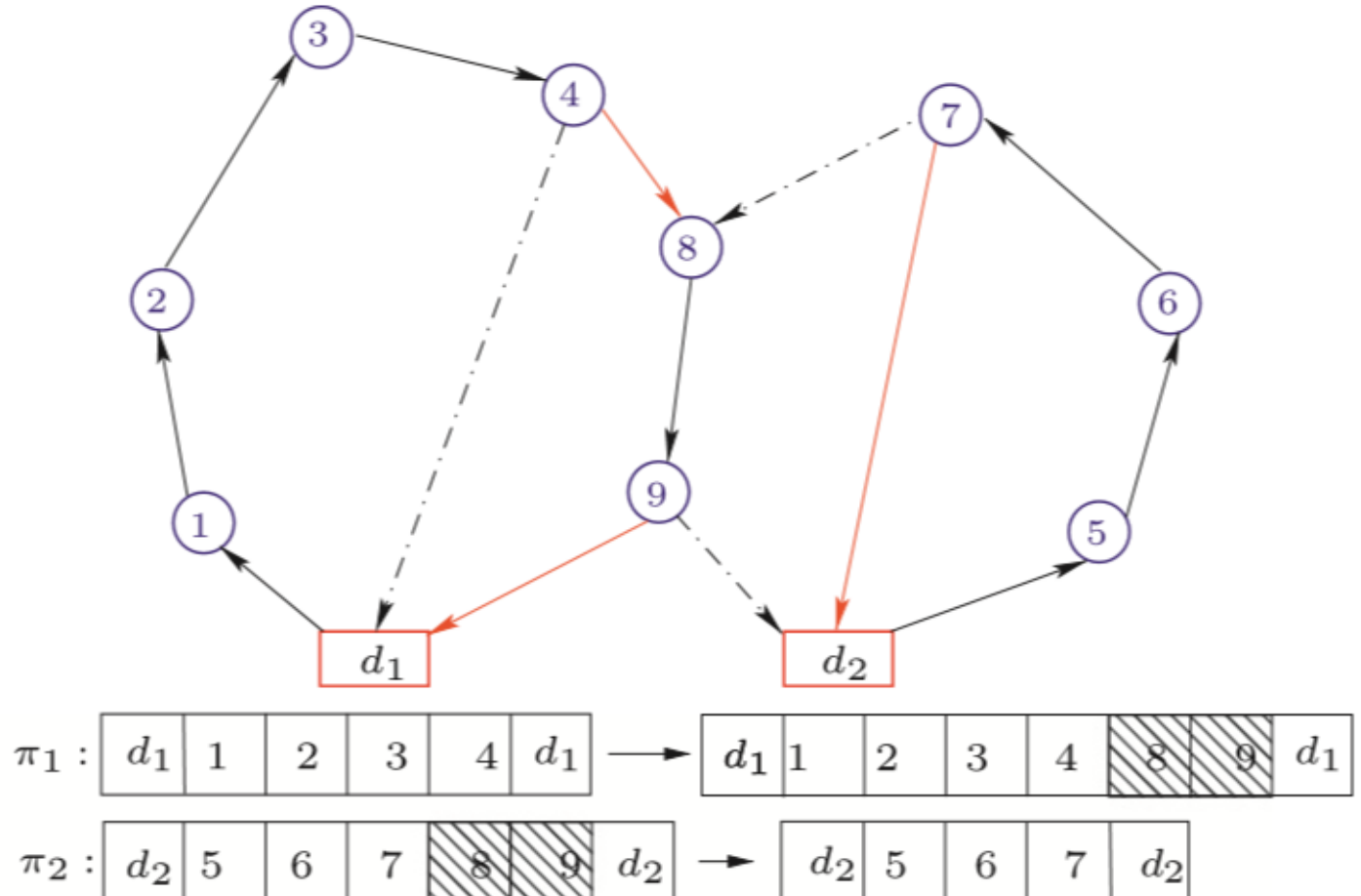
Jarboui et al. (2013)



Operadores para geração de vizinhança

Or-opt (entre rotas)

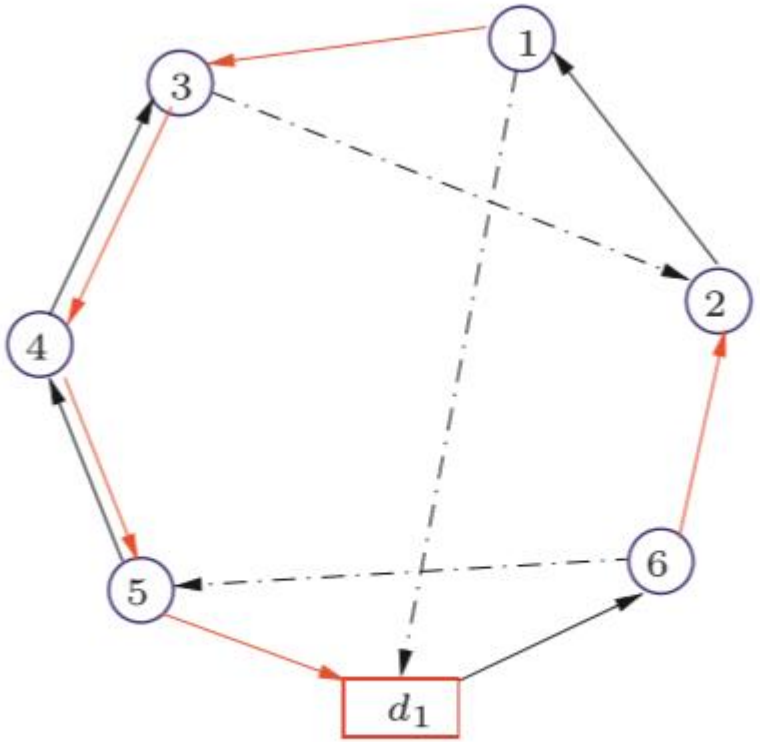
Jarboui et al. (2013)

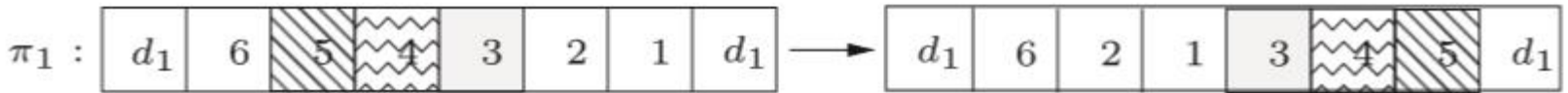


Operadores para geração de vizinhança

Or-opt invertido (mesma rota)

Jarboui et al. (2013)

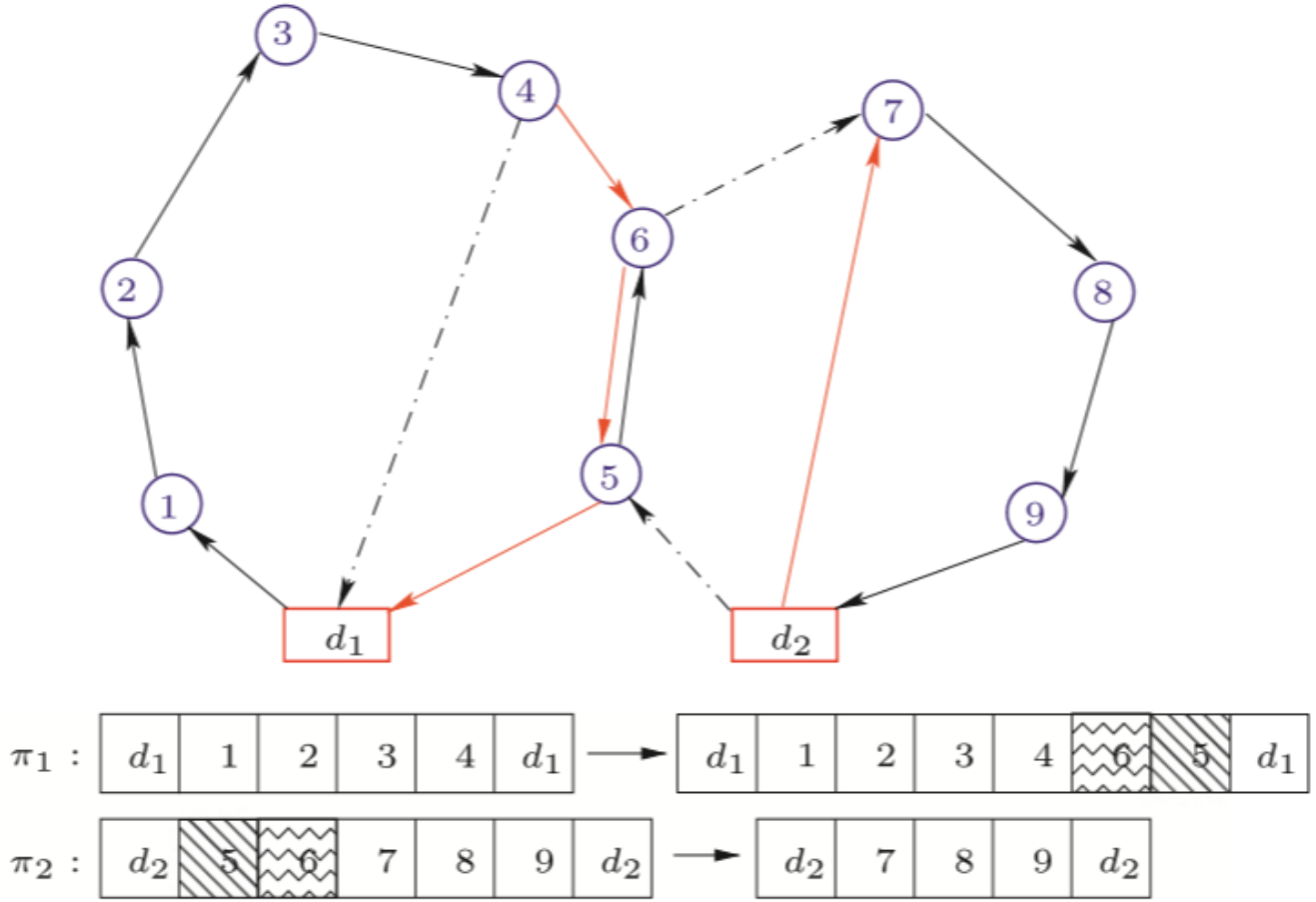




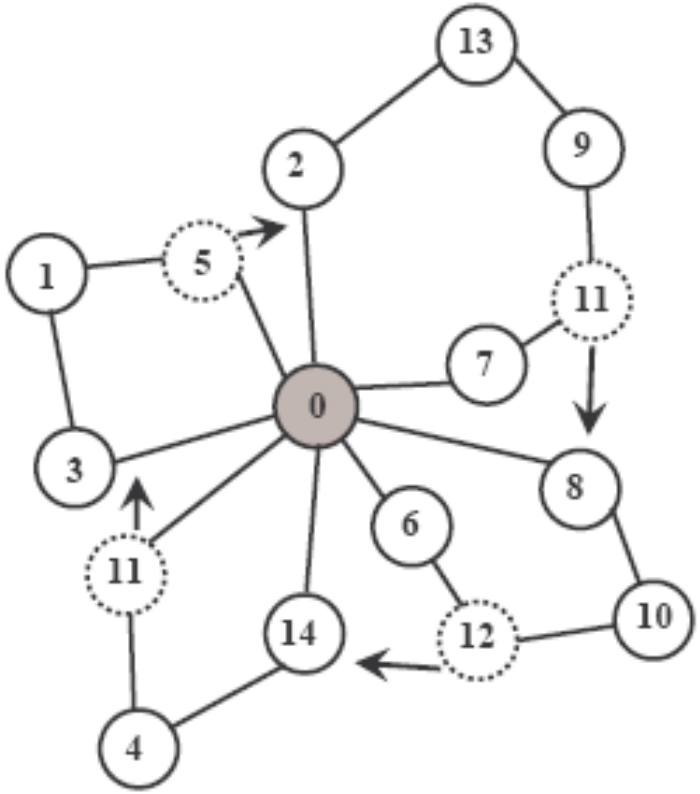
Operadores para geração de vizinhança

Or-opt invertido (entre rotas)

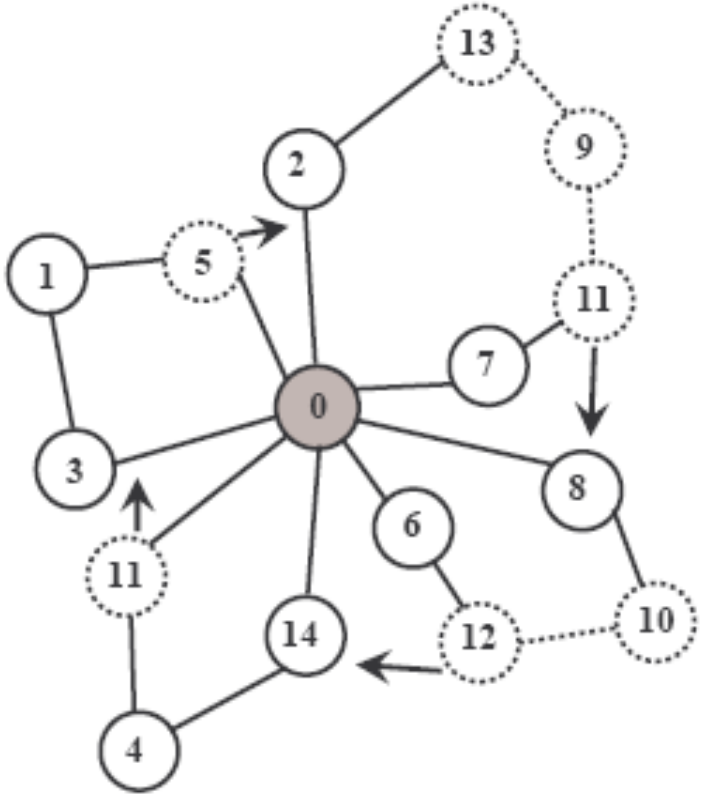
Jarboui et al. (2013)



Operadores para geração de vizinhança



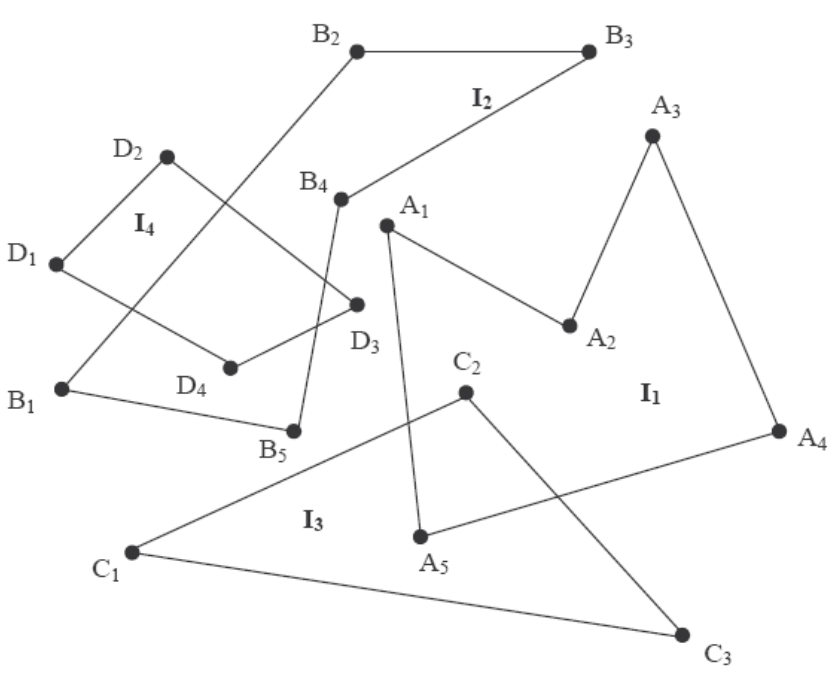
k=1

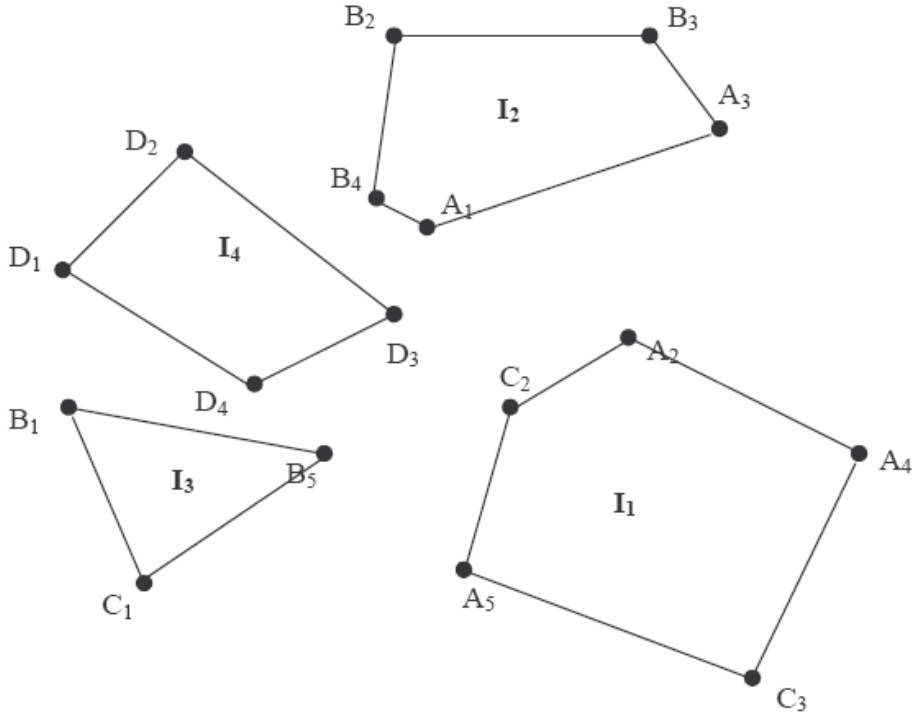


k genérico

Operadores para geração de vizinhança

Vizinhança: "*b*-cyclic, *k*-transfer scheme" - *k* clientes <u>consecutivos</u> de cada uma das *b* rotas são transferidas para a próxima rota





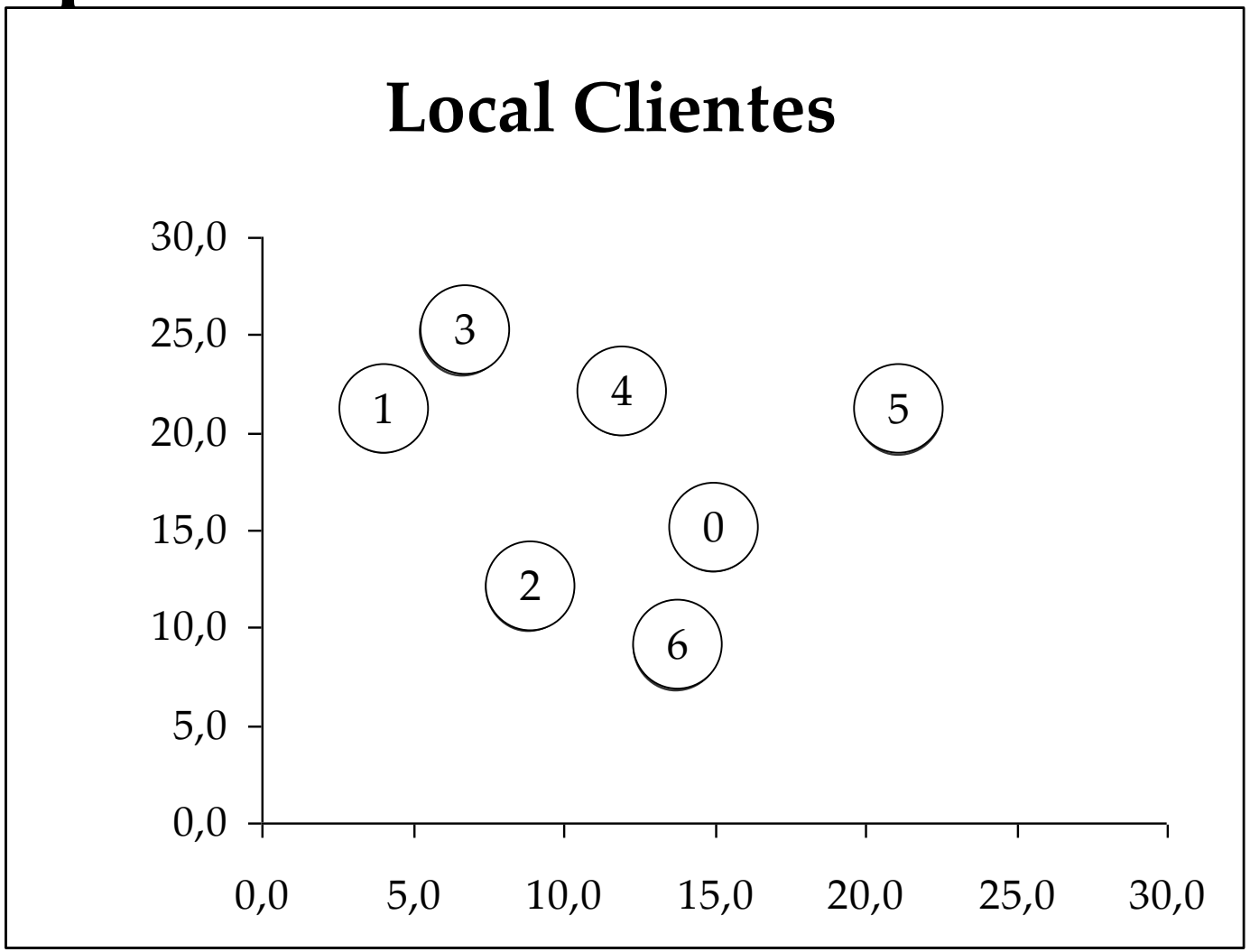
Algumas referências

- Heurística de Busca
  - Heurísticas desenvolvidas para problemas de roteirização e programação de veículos: 2-Opt (Lin, 1965), “2-Opt*” (Potvin; Rousseau, 1995), “relocate, exchange, cross” (Savelsbergh, 1992), “CROSS” (Taillard et al., 1997) “GENI” (Gendreau et al., 1992) e “cyclic transfer” (Thompson; Orlin, 1993)

Método de Busca *2-Opt*

- Este método consiste em identificar 2 arcos não-adjacentes que serão eliminados da rede, para que novos arcos sejam religados, com o objetivo de reduzir a distância total;
- É necessário partir de uma solução inicial viável (fornecido por alguma heurística);
- O método encerra quando não houver mais arcos que permitam a redução de distância;

Método de Busca *2-Opt*
Exemplo - PCV

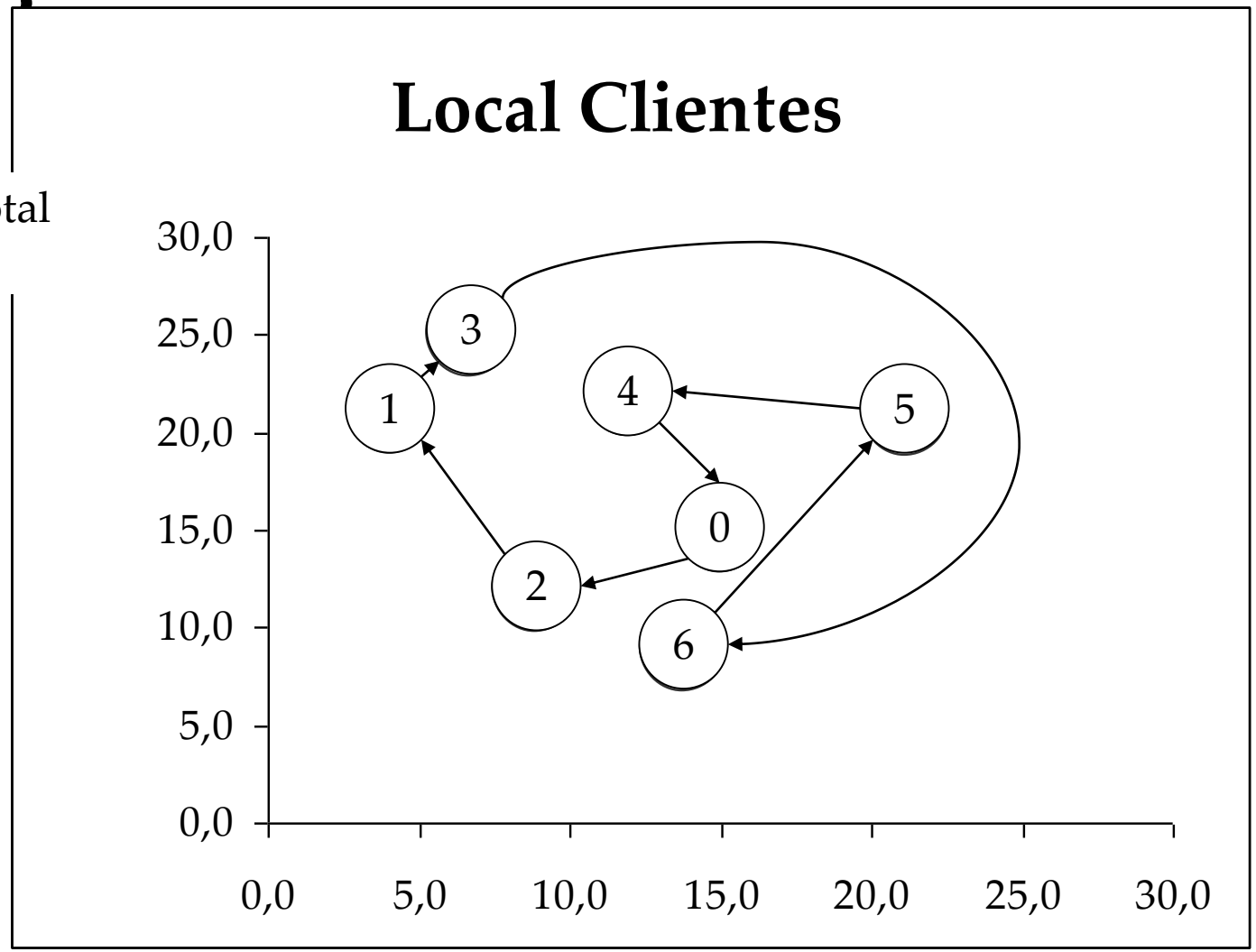


Método de Busca *2-Opt*
Exemplo - PCV

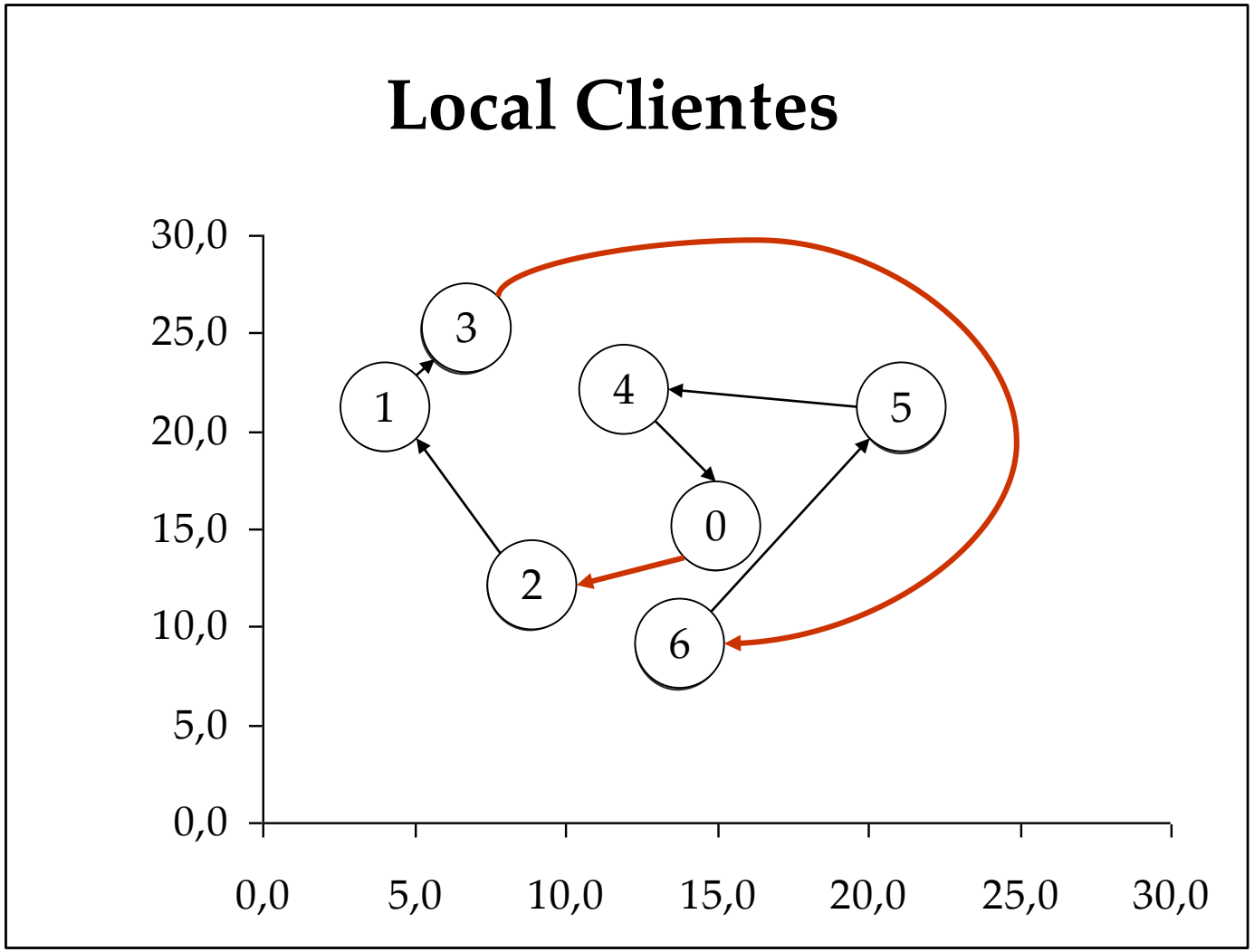
	0	**1**	**2**	**3**	**4**	**5**	**6**
0	0,0	12,4	6,7	12,9	7,6	8,6	6,2
1	12,4	0,0	10,2	4,6	7,9	17,1	15,5
2	6,7	10,2	0,0	13,1	10,4	15,2	5,8
3	12,9	4,6	13,1	0,0	6,0	15,0	17,6
4	7,6	7,9	10,4	6,0	0,0	9,2	13,2
5	8,6	17,1	15,2	15,0	9,2	0,0	14,1
6	6,2	15,5	5,8	17,6	13,2	14,1	0,0

Método de Melhoria *2-Opt*
Solução Inicial

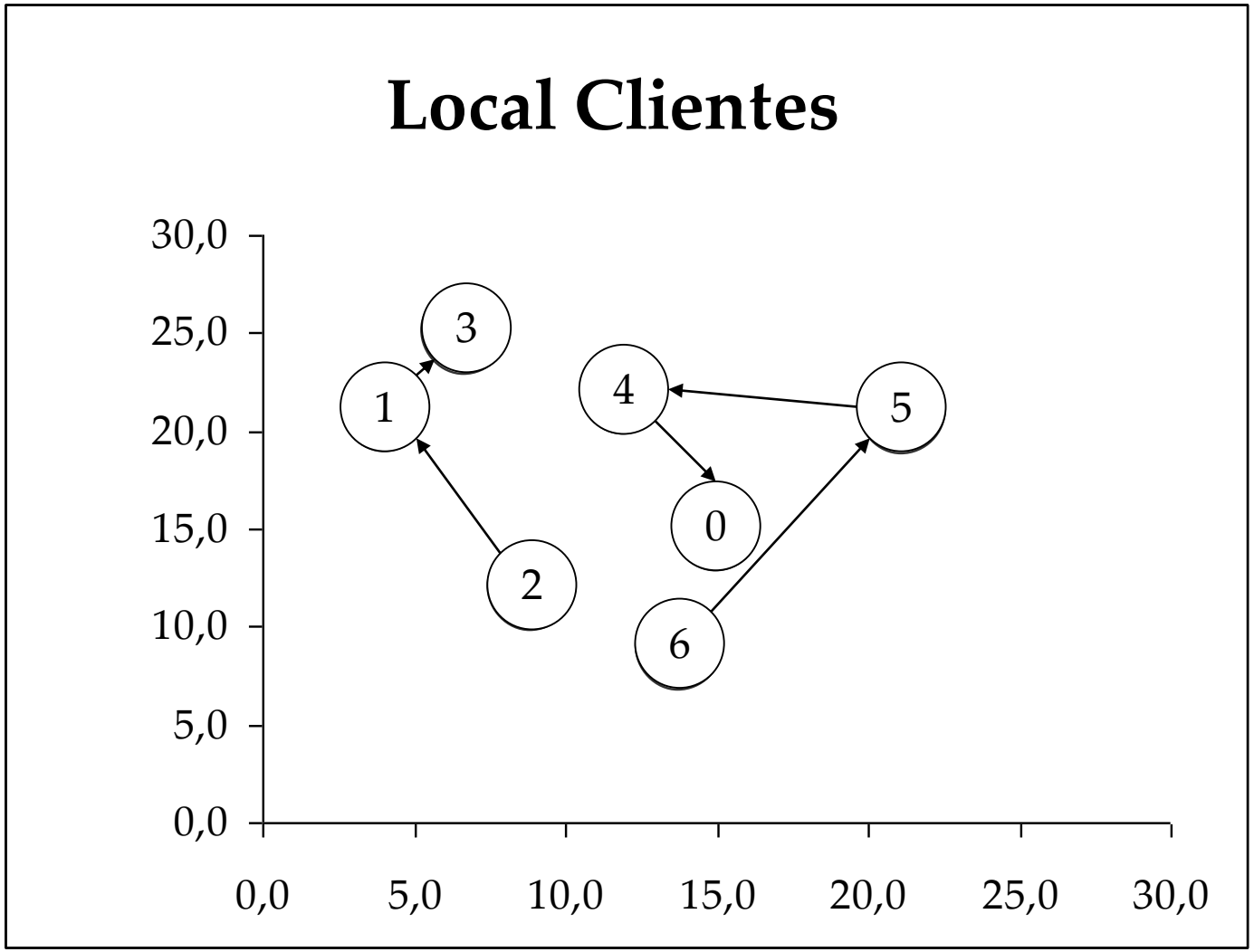
Distância Total
= 70,0 km



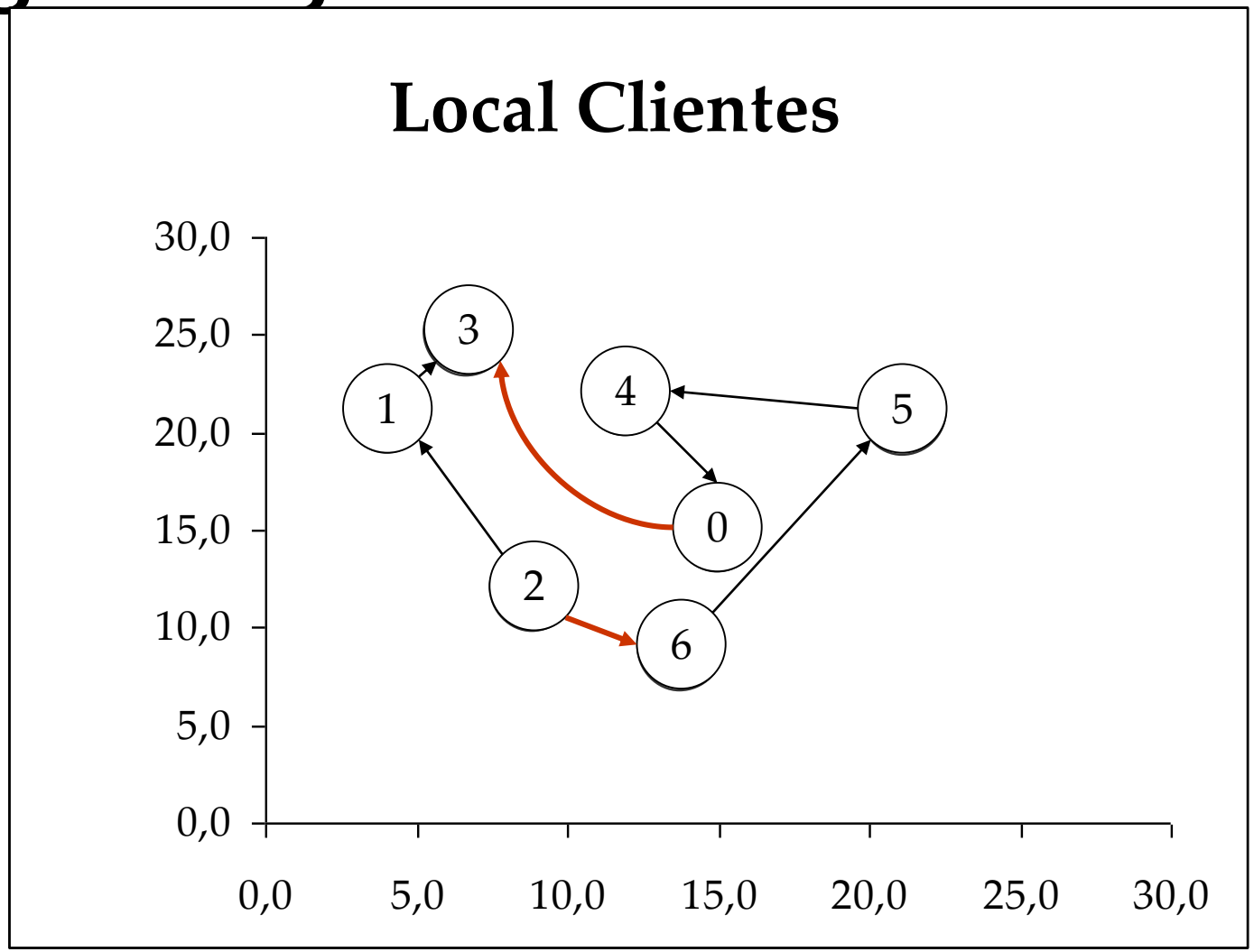
Método de Busca *2-Opt*
Identificar arcos



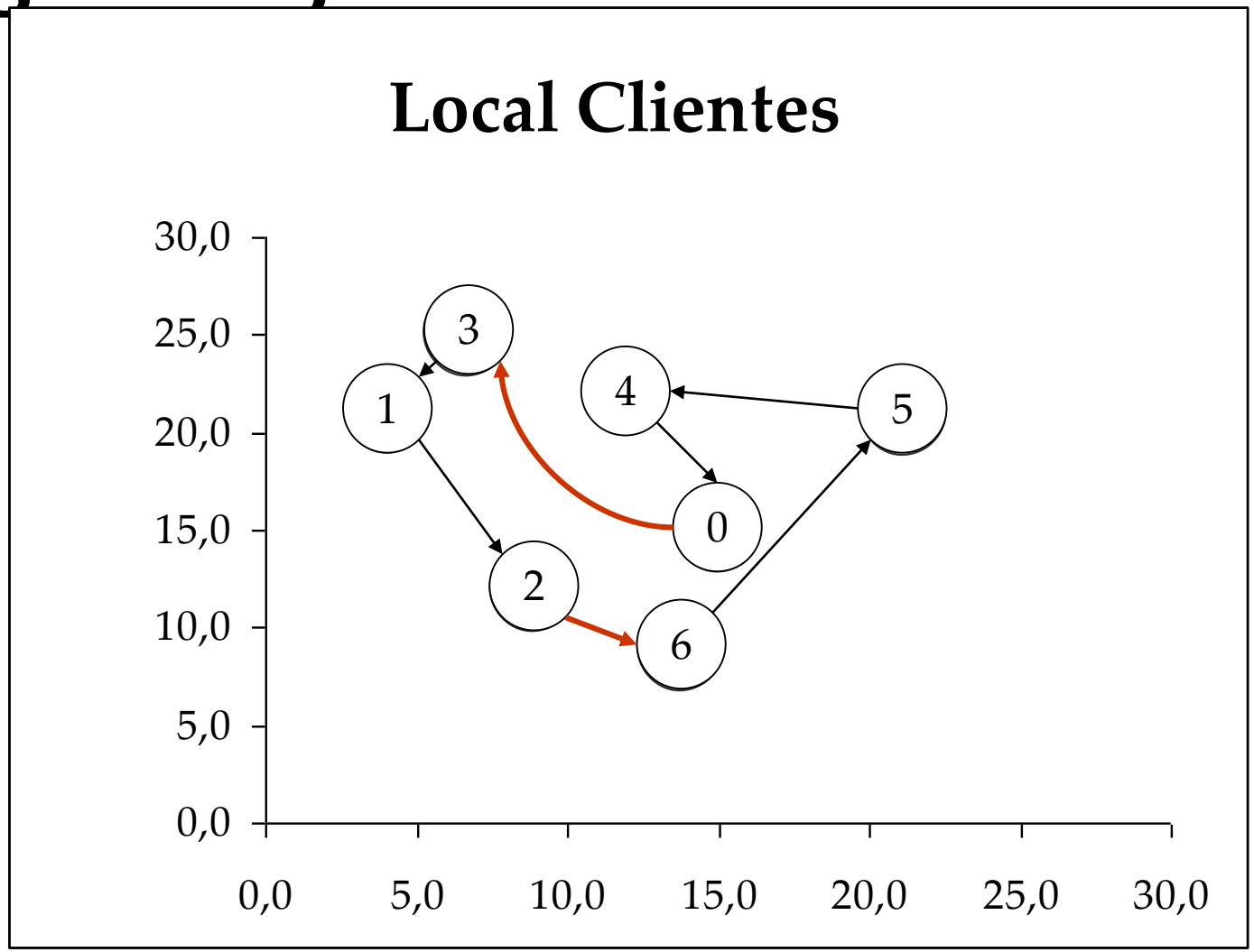
Método de Busca *2-Opt*
Eliminar arcos



Método de Busca *2-Opt*
Religar & Ajustar o Sentido dos Arcos

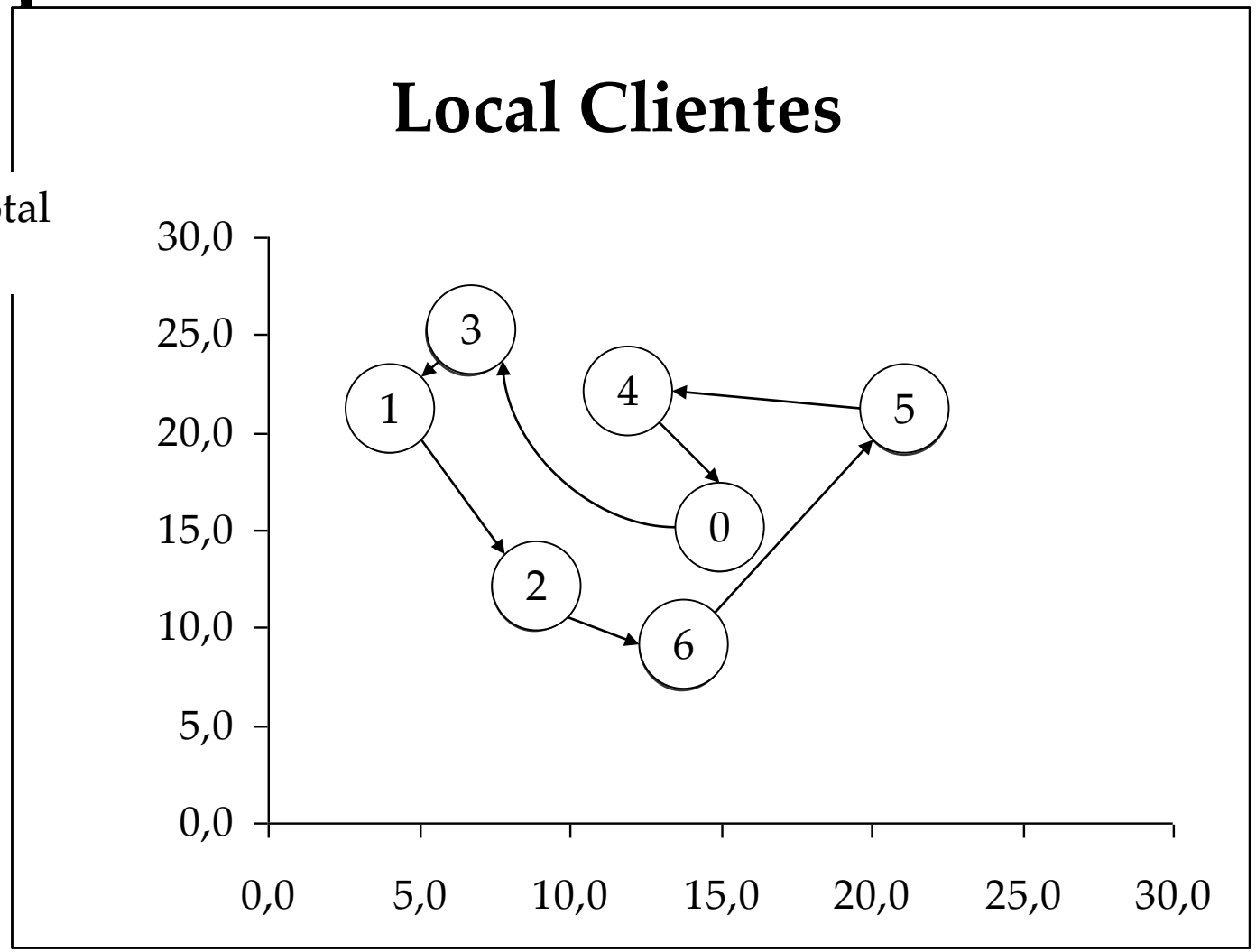


Método de Busca *2-Opt*
Religar & Ajustar o Sentido dos Arcos

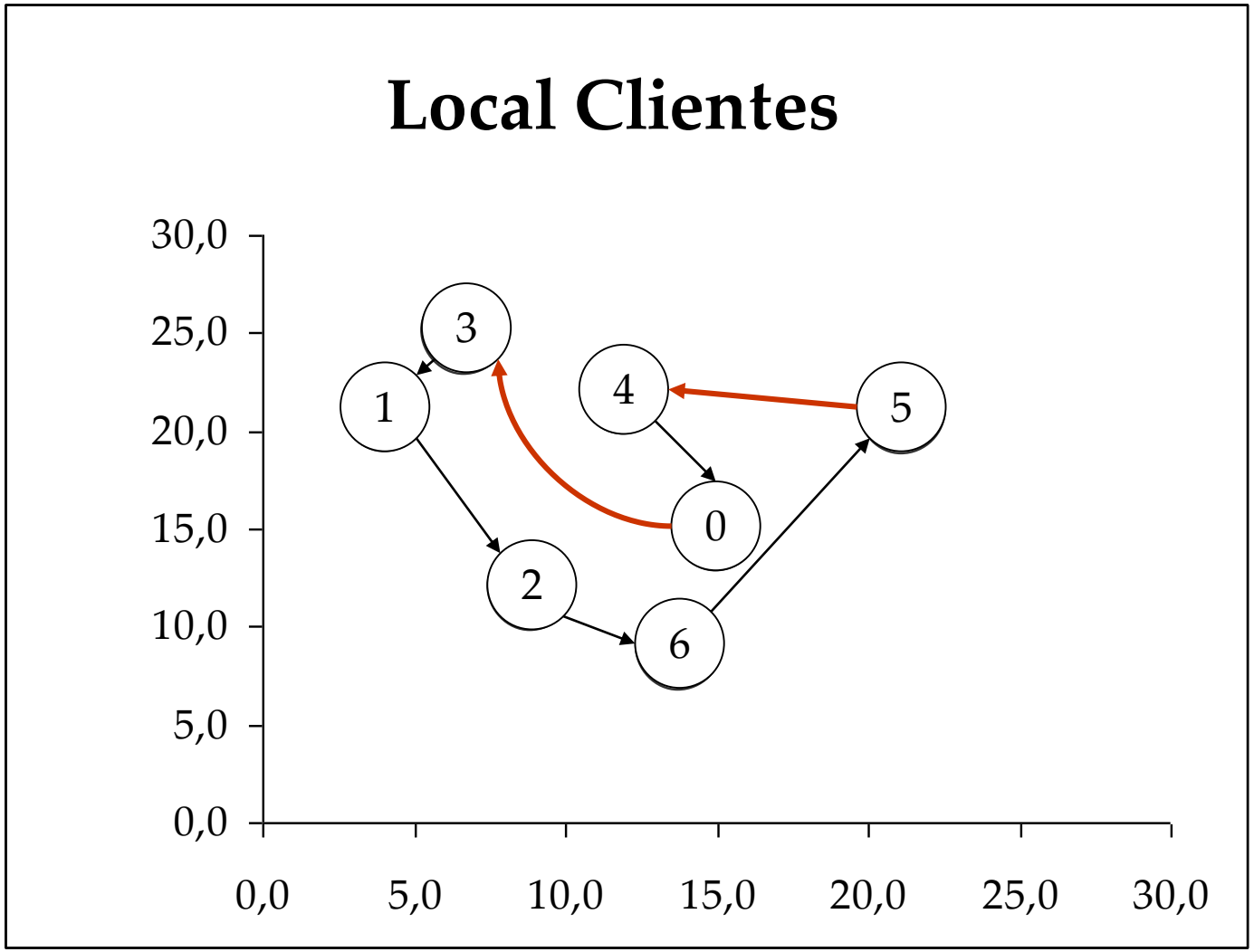


Método de Busca *2-Opt*
Solução 1

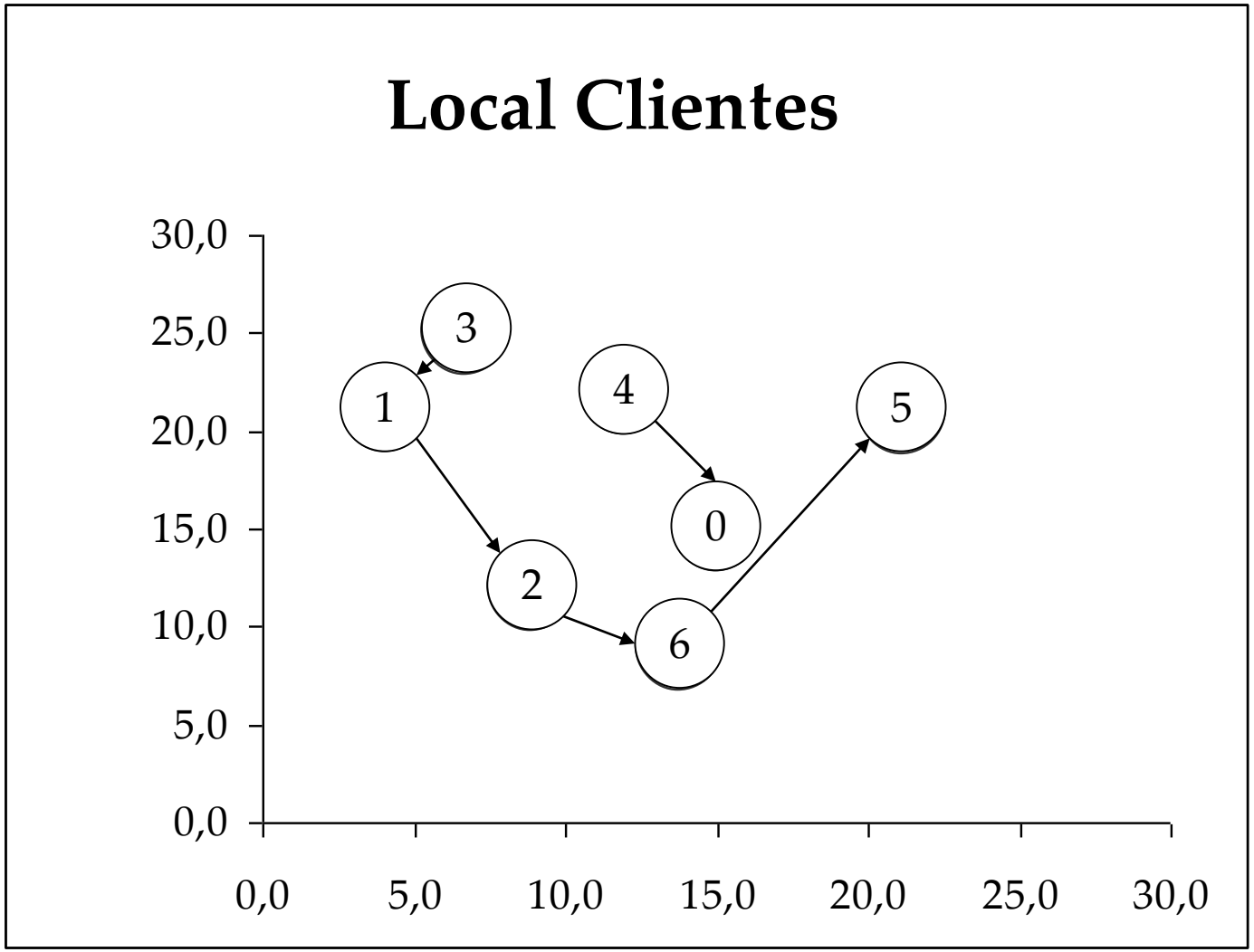
Distância Total
= 64,4 km



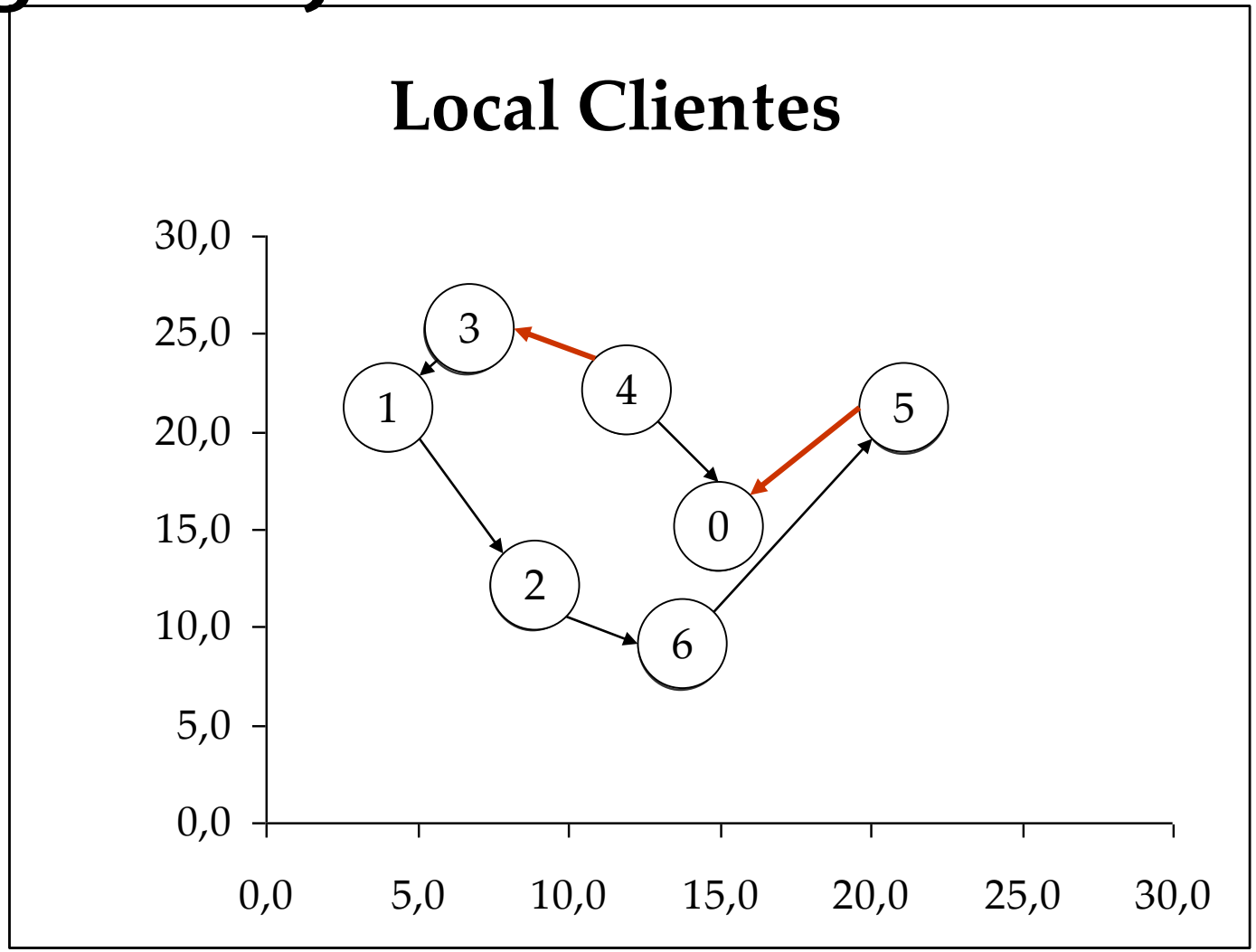
Método de Busca *2-Opt*
Identificar Arcos



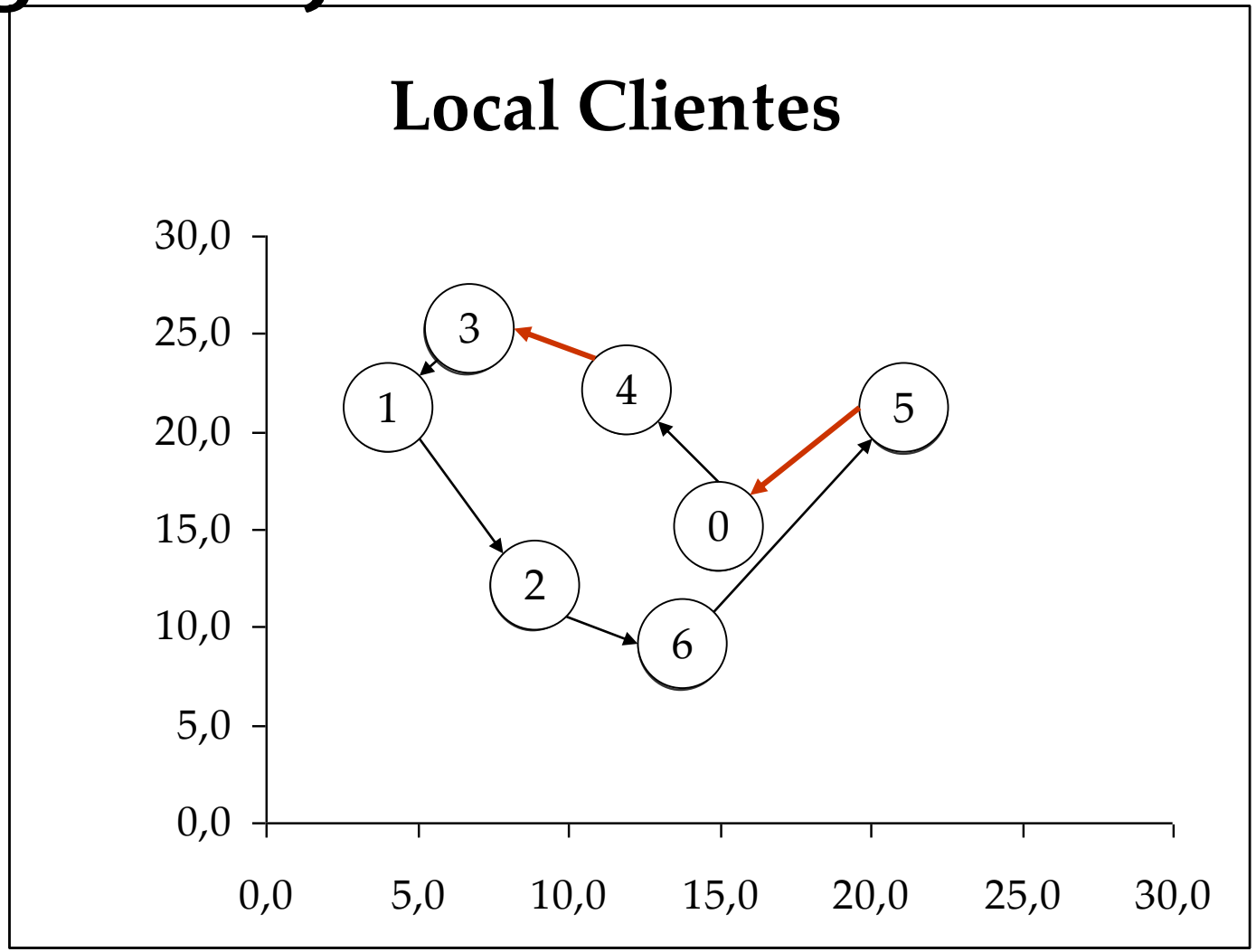
Método de Busca *2-Opt*
Eliminar Arcos



Método de Busca *2-Opt*
Religar & Ajustar o Sentido dos Arcos

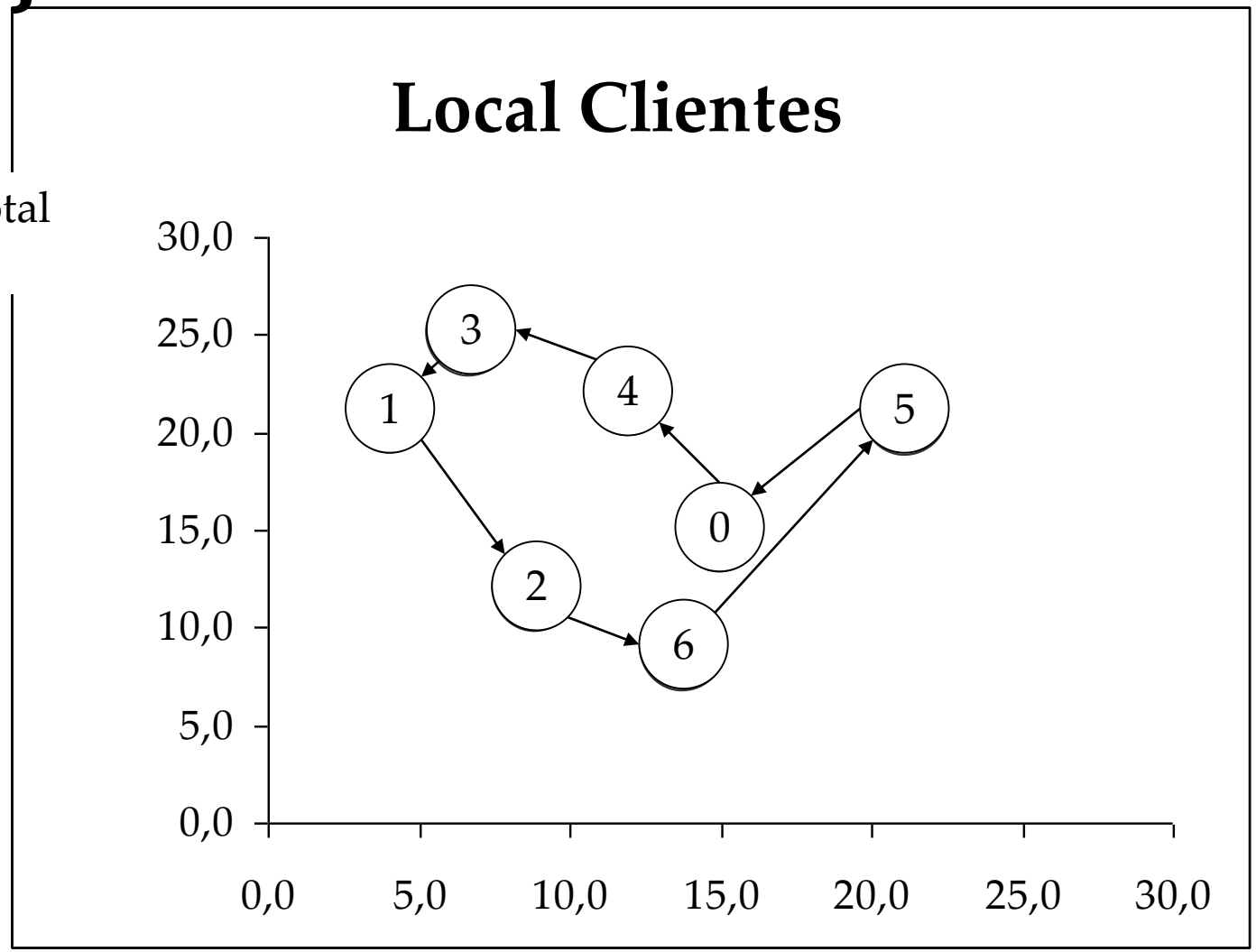


Método de Busca *2-Opt*
Religar & Ajustar o Sentido dos Arcos

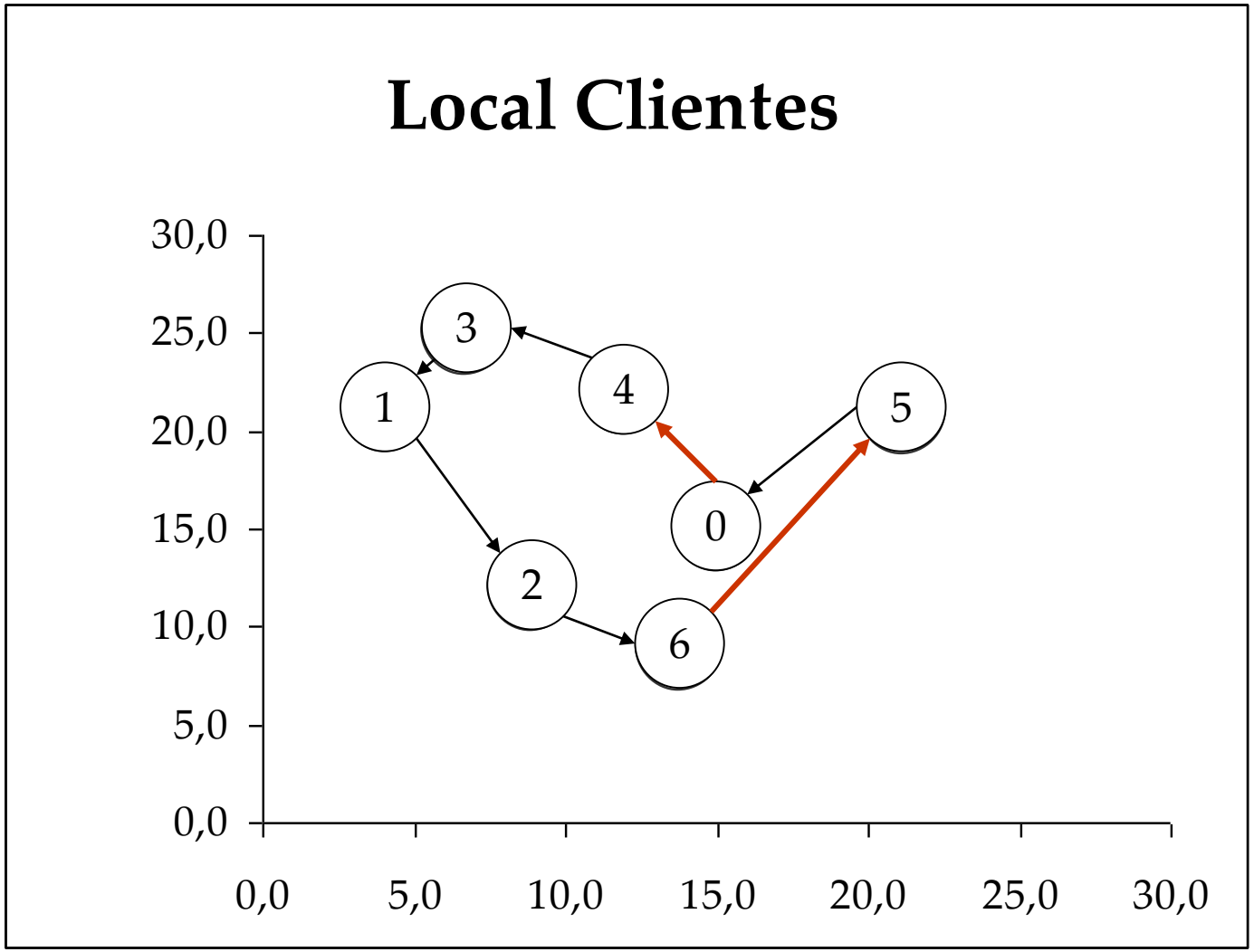


Método de Busca *2-Opt*
Solução 2

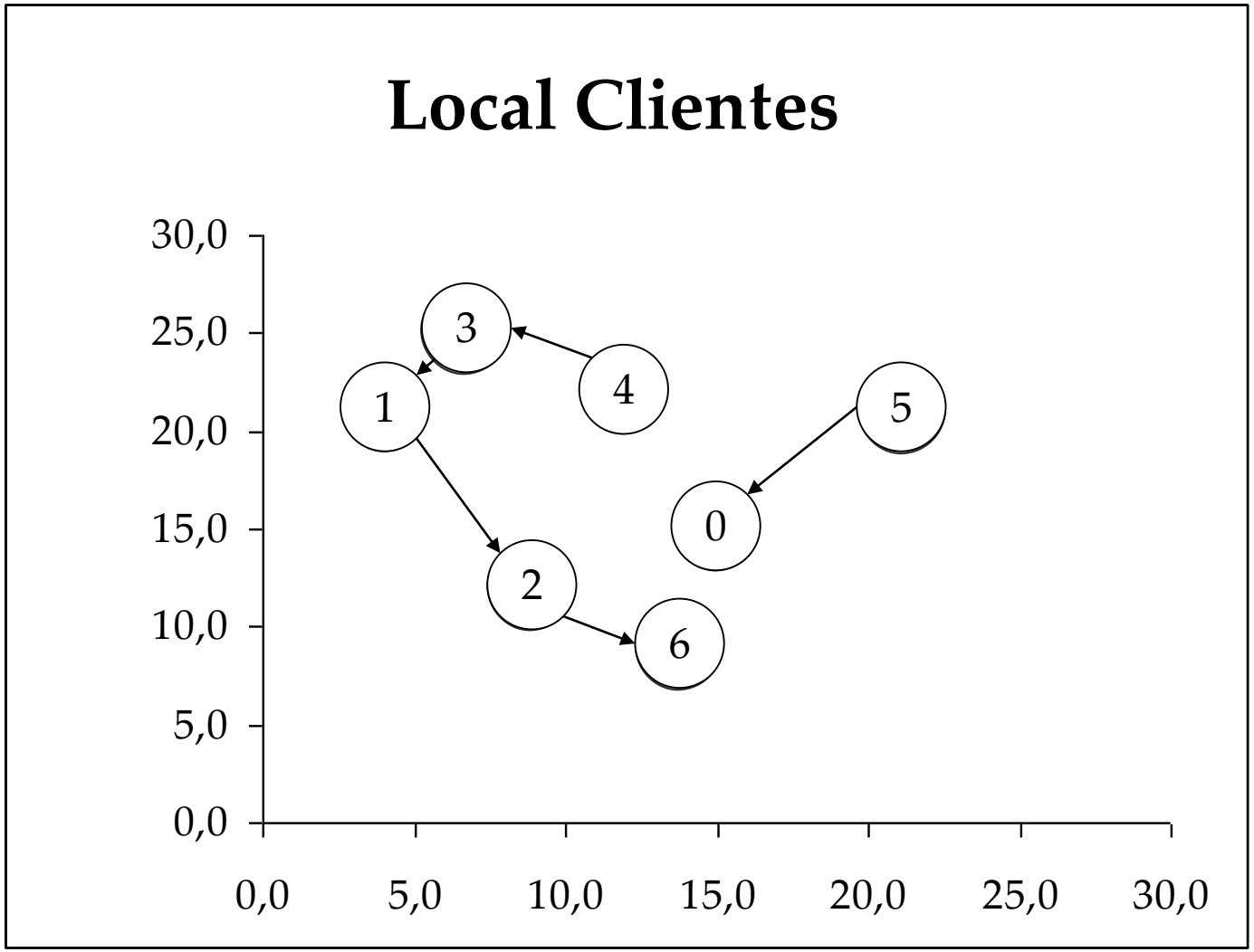
Distância Total
= 56,9 km



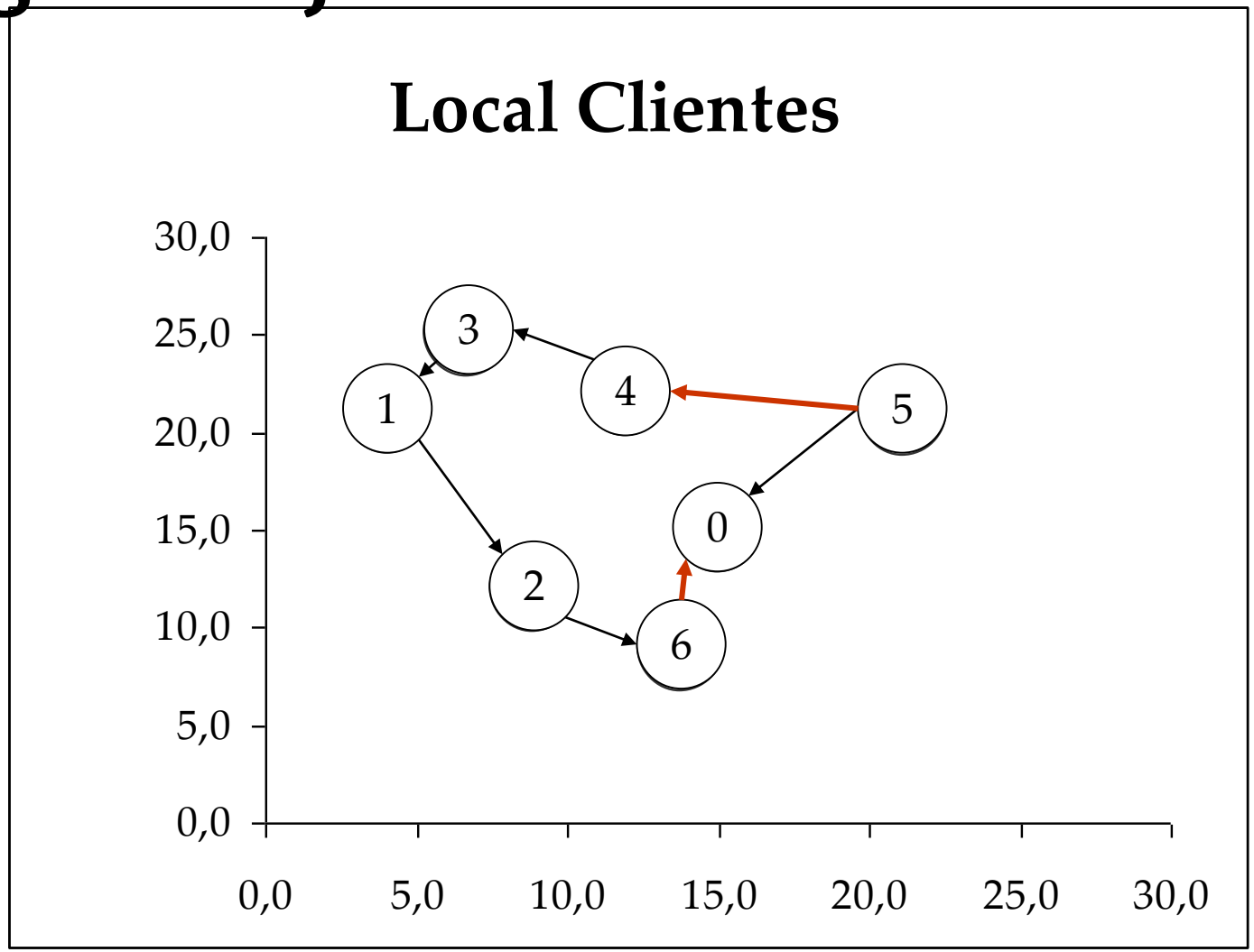
Método de Busca *2-Opt*
Identificar Arcos



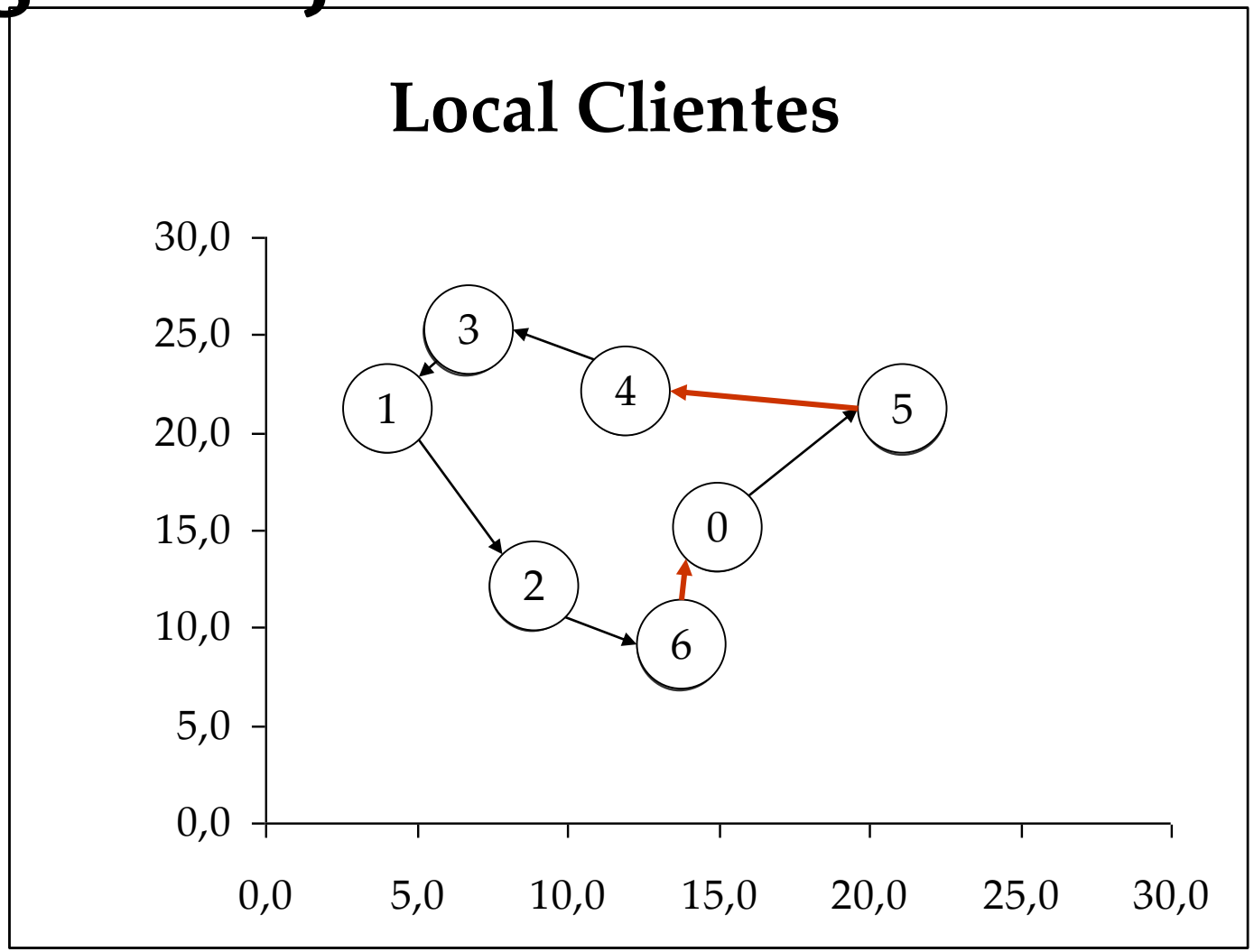
Método de Busca *2-Opt*
Eliminar Arcos



Método de Busca *2-Opt*
Religar & Ajustar o Sentido dos Arcos



Método de Busca *2-Opt*
Religar & Ajustar o Sentido dos Arcos



Método de Busca *2-Opt*
Solução 3

Distância Total
= 50,6 km